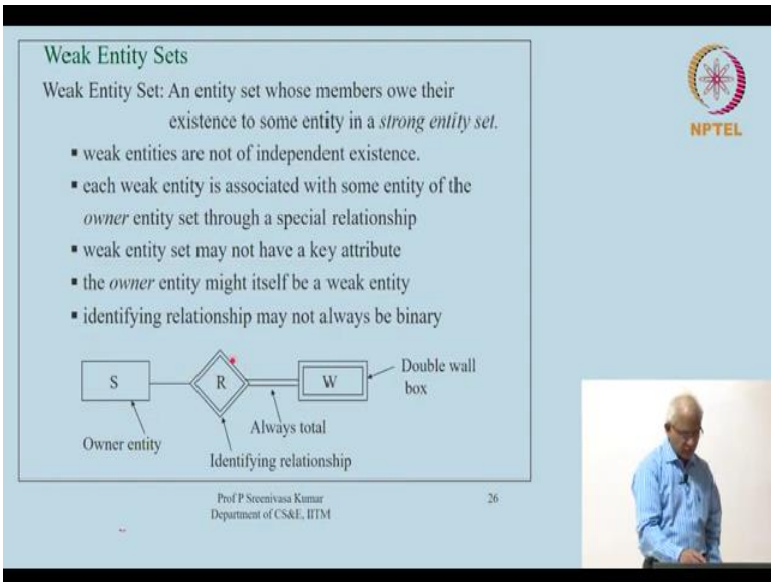
Database Systems
Prof. Sreenivasa Kumar
Computer Science and Engineering Department
Indian Institute of Technology-Madras

Lecture-06
Modelling Weak Entities and Design Choices

Now we have introduced this notion of entities that is what we started of with right entities.

(Refer Slide Time: 00:25)



And entity is something which we are concerned about it is an important thing that we would like to model in the information system. And then entities had attributes and then they participate in relationships etc. Now, in practice we also encounter slightly different kinds of entities. So, these are called weak entities. The difference between this.

And the entities that we have been discussing so far which now can we actually call a strong entities is that the existence as far as this information system is concerned, the existence of these entities in this weak entity set you know depends on the existence of some other entity unless that entity exists, there is no point talking about this entity which we are calling it as a weak entity.

So, these weak entities are not have independent existence, their existence is owed to the existence of some other entity possibly in a strong entity set okay. So each of these weak entities is associated with some entity of we call them as a owner entity through a special relationship, okay. Now when we discuss examples it will become very clear.

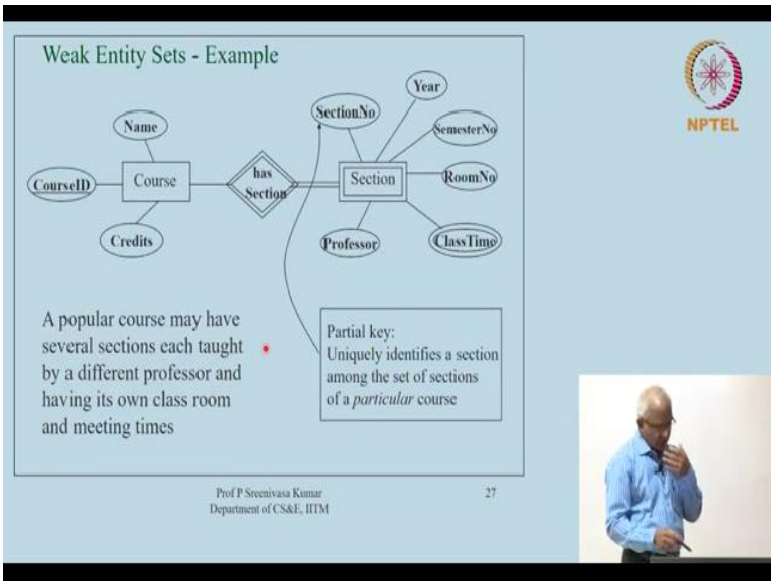
But basically what we are saying is that there are certain kinds of entities whose existence in some sense depends on the existence of other entities and we call them call the other entities as the owner entity, because the owner entity exists, this other entity, which we are calling as weak entity becomes important for us.

Now, let me give you an example. We will see couple of examples. So, the weak entity said may actually not have a key attribute at all. It may have what is called a partial key, we will talk about it and the owner entity might itself be a weak entity okay. It is possible that the owner entity, what is the owner entity, a owner entity is something you know, which kind of owns this weak entity, that itself can be weak in which case there is a chain of these weak entities.

And so ultimately of course, it has to be owned by some strong entity. Now, okay here is the diagrammatic notation for that. So, the weak entity is shown with a double wall box we used to use normal rectangular boxes earlier for entities, now weak entity will be shown with a double wall box and the name of the weak entity will be written here and it is there is always a special relationship that connects this weak entity to the owner entity okay.

So, this kind of this relationship is also called the identifying relationship and this weak entity you know always participates totally in this identifying relationship, because this weak entity cannot exist unless it is related to some strong entity owner entity. So, this kind of situations arise and in order to capture their you know meaning in the form in the domain. We like to I mean we use this notion or concept of weak entities.

(Refer Slide Time: 04:38)



So, let me give you an example, let us say a popular courses there. Nowadays, one of the popular courses is machine learning, right. So, you offer multiple sections of the machine learning people will enroll for that, I hear that enrollment for machine learning, you know crosses 100 or something like that okay. So let us say there is some popular course. And so we decided to have sections for the course.

So what is the section, a section is an offering of that course. And that is where actually students enroll into these sections, each section is taught by a professor, maybe a different professors, we have multiple sections, A B C D, A B C sections, each of them is taught by a different professor and each of its has its own classroom and meeting times right. So it has all these things.

And it is students enroll for these sections. Right now, but you can see that why section is a weak entity is that unless the course exists the question of having a section does not arise right. So we have a course and it is popular and so we would like to you know and even otherwise, whenever you know, a course is being offered in multiple, you know, years and things like that, we can always call it an edition of the course right.

So, for the 2019 database systems course your read, you all kind of enrolled it, enrolled into. So, this edition exists because the course exists right, course is a strong entity and section is a weak entity sometimes. So there will be multiple sections associated with this, the course and so, we

model this as a weak entity, mainly because it is also said existence, to the existence of the strong entity called this database systems courses, and whatever course it is.

And we call this has section relation as the identifying relation and this is called a weak entity and this is called its corresponding owner entity and we show the arrangement like this. Now, the sections, all sections corresponding to one particular course, you know, like CS 1100 like the introduction to computing, there are usually 4 sections for that course, right.

So, we have to identify those sections is this section A, section B, section C and sections D. So, we give some section number 1 2 3 or some letter A B C D and that will be identifying the actual know the sections of that particular course. So, such a thing in attribute like section number, or section name, a section letter or something like that which has this property that it will identify the exact section.

Among the sections that are associated with a particular course, is what we are calling as partial key. It is not a key, because if you just consider all possible sections being offered by the department, and then say that, okay, this student is enrolled in section A, and it does not make sense, right. What section A are you talking about, CS 110 has section A, mission learning also has a section A and something else also has a section A.

So what section A are you talking about. So this section A by itself, you know, does not identify any particular section. And so that is why it is called a partial key. So the course whatever the course ID along with this section number together only actually identifies the section uniquely okay because all these sections are associated with that course. So, the course ID or whatever is the key for that particular strong in the owner entity.

Along with this partial entity will uniquely identify this particular section. In general, that is how it is. So it might have a some attribute, you know, which might act like a key, but it is not necessary that weak entities have a key, that is the point we are trying to make. In the in the book by Elmasri and Navathe they give an example of dependence of an employee. So, the

organization is interested in keeping track of the information about the dependence of employees as long as the employee exists in their organization.

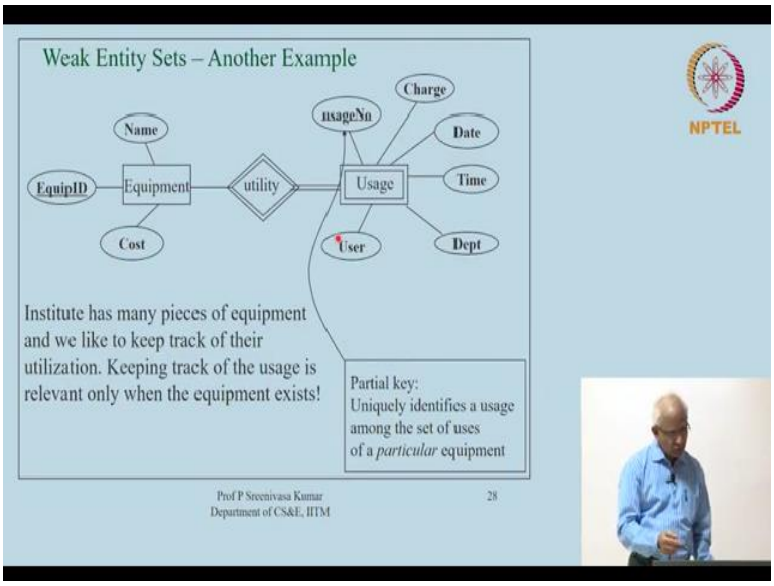
If that particular person leaves the organization then there is no point keeping track of the information about his or her dependents. So, the dependence of a particular employee or best model as weak entities in the organization, because they are basically interested in those dependents, because they have hired that particular employee right. So, those are dependents will become weak entities owned by the corresponding employee.

So, in this case you can see that those dependents or anyway a person's. And so, they might actually have some identifying attribute like social security number or aadhar number or something like that, which is an identifying attribute right and it is a key attribute. It identifies the person uniquely. But that is besides the point that as far as the information system is concerned it is not necessary that all these weak entities that we are we may be interested in may have a key.

So, but what we have guaranteed is that we should introduce some attribute which can uniquely identify the all the sections or all the weak entities associated with it strong entity and usually we can find such a attribute and that is what is what we call a partial key okay. So, is this idea of a weak entity clear. Essentially it is a we are saying that there are situations where we like to keep track of certain entities.

But only thing is that those the existence of those entities depends on the existence of some other entities which we are interested in. Now, let me give you another example.

(Refer Slide Time: 11:54)



The institute has many pieces of you know equipment which are used for scientific experimentation and things like that. And we would like to keep track of their utilization, how, who is using them and when they are using it and things like that. So, usage of a particular equipment, you know is best model as a weak entity. Unless the equipment exists there is no point talking about the usage of that equipment, right.

The usage is modeled as a weak entity, and it is connected to the equipment strong entity, owner entity through this identifying relationship called utility. And then it has its own attributes, what is the usage number, something which uniquely identifies the usages of a particular equipment and then who has used it and which department that particular person belongs to, when was it used date, time and what are the charges that you may have.

We may internally have some arrangement of charging the person for using that equipment. And you know things like that. So, again this is best model as a as a weak entity and usage number serves as the some partial key here which will identify the usages of a particular equipment. Typically this kind of information will be maintained in a small register you know associated with that.

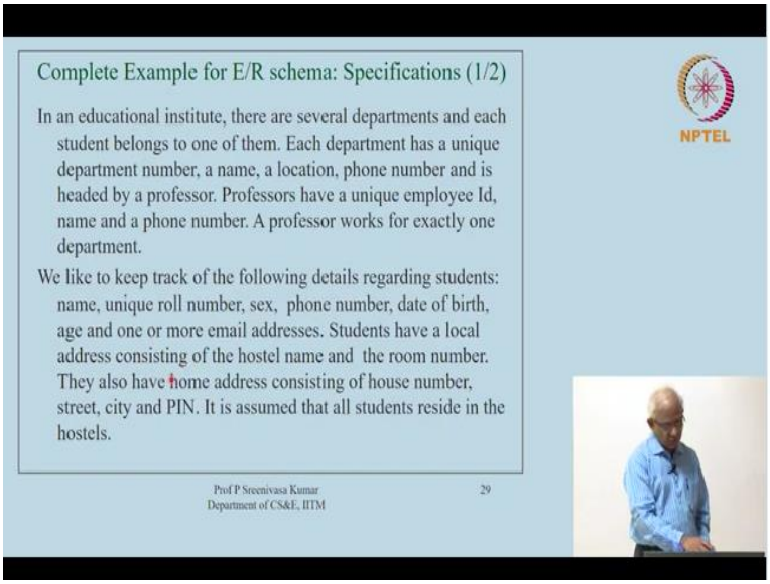
So, you go to a photocopier then they will say okay, you write your name and then you know use it and they at least know that okay, it has been used some so many pages have been copied and

things like that right okay. So with this the principal constructs that, that are available the conceptual tools that are available in entity relationship, model have you actually covered. So, let me summarize that.

Basically, the entity relationship model is a data model that means it is a collection of conceptual tools to describe the database at the conceptual level. And it has these notions of entities, attributes for entities and various kinds of attributes, you are seeing them and relationships, relationship types, and various kinds of you know, one to one, one to many, different kinds of relationships and then the notion of weak entities.

All these things together give us a bunch of tools to kind of capture the domain information and represented in a diagram.

(Refer Slide Time: 15:03)



So I will, what I will do next is to look at you know, typical collection, how do we write down the specifications, you know, in plain English and then later actually translate that into a bigger diagram entity, ER model. So this particular example is what I am going to use in later on also when we discuss relational data models, etc. So let us go through that.

So, some of many of these concepts have actually been introducing as examples as well, as I am discussing these various notions. So let us capture all of them together and put them at one place

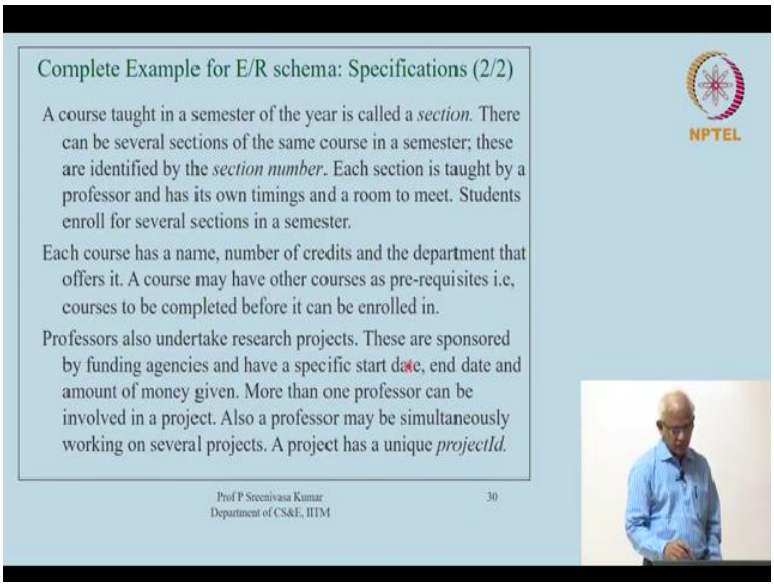
as a complete example. So education institution, there are several departments and each of and each student belongs to one of them. And the department has a unique department number, a name and a location, phone number and is carried by a professor.

Professors have a unique employer and name and phone number and professor works for exactly one department. This is how we can jot down the domain information and so that we can later translate that into an ER diagram. Later on in an assignment, I would actually like you to do this kind of an exercise where you know you, you choose a domain and then jot down the main facts that you want to reflect in the domain model.

And then actually translate that into an ER model and submit that. So we like to keep track of the following details regarding students name, the roll number, unique roll number, sex, phone number, date of birth, age, and one or more email addresses. Students have a local address consisting of hostel name and room number. They also have a home address consisting of home number, city , street pin, it is assumed that all students resides in hostels.

Now you actually wonder if a particular attribute is not applicable for a particular entity, what should it be its value okay. We typically use what is called a null value for that okay.

(Refer Slide Time: 17:33)

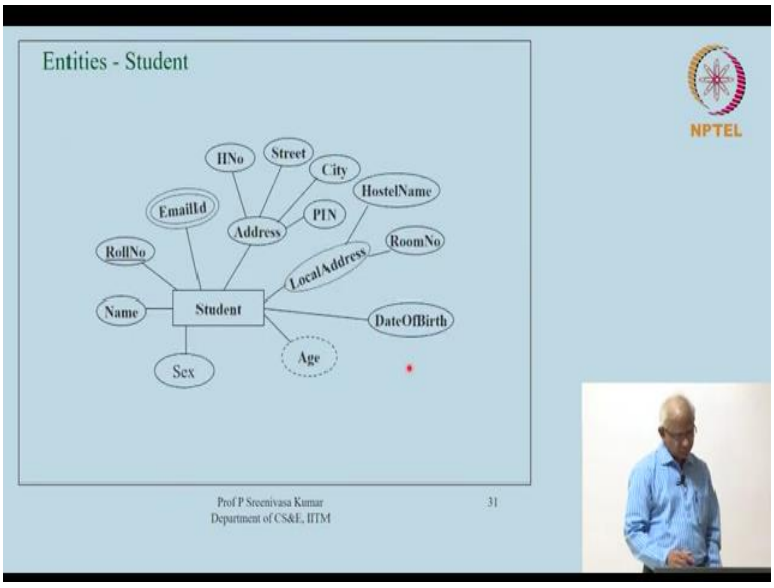


We will meet that again later on also okay, a courses taught in a semester of the year, and it is called a section okay. So whatever courses are not directly taught, it is a section of the courses that are taught. So when it is a taught in a semester of the year, it is called a section. There can be several sections of the same course in a semester. These are identified by the section number. And each section is taught by a professor and has its own timings and a room to meet.

Students enroll for several sections in a semester, in a given semester students enroll for several sections. And each course has a name, number of credits and the department that offers it. There is a owner department for it. A course may have other courses as prerequisites, that is the courses to be completed before this particular course can be enrolling. And then professors also undertake research projects.

And these are, you know, sponsored by the funding agencies and have a specific start date, end date and amount of money given to them. And then more than one professor can be involved in a project. Also, professor maybe simultaneously working on several projects, a project has a unique project ID. So, this is how we gather together all the facts that we want to model and then start putting them all together in a diagrammatic notation the ER model notation. And then have discussions with the end users.

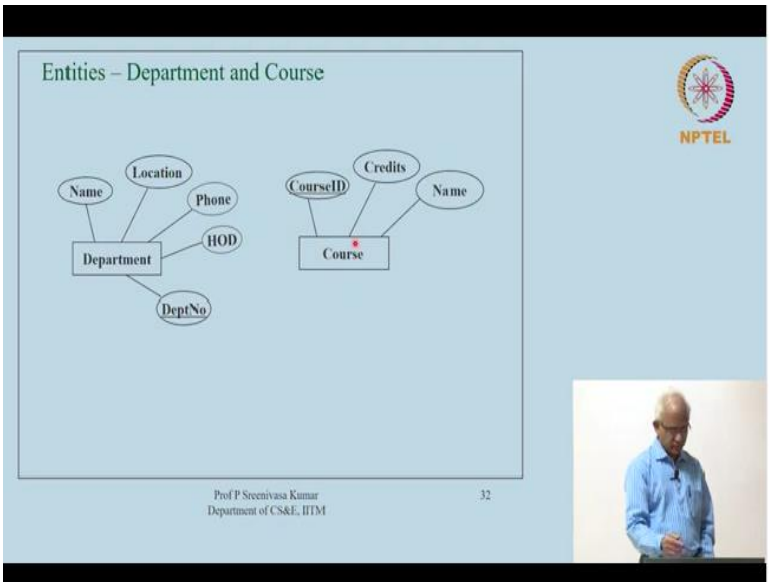
(Refer Slide Time: 19:26)



So here is how student entities model, so we have seen it already. So let us just see that the name is just you know as per the writing there, we ignored that name has any other components, we just named it as name, roll number is unique. So we underline the key attribute because it is value can we use to uniquely identify the student identity. There are one or more email id so it is a double this one ellipse.

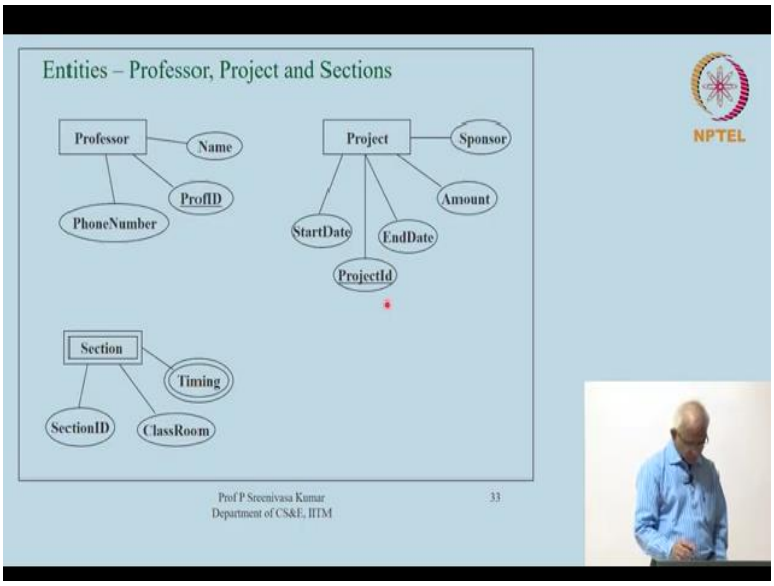
Then we have address, addresses has is a composite attribute because it has house number street city and pin and the local address is also a composite attribute because it has a hostile name and room, we are interested in date of birth and age is a derived attribute and we are interested in the gender of the student. So, these are the various attributes. So, we typically draw like this.

(Refer Slide Time: 20:38)



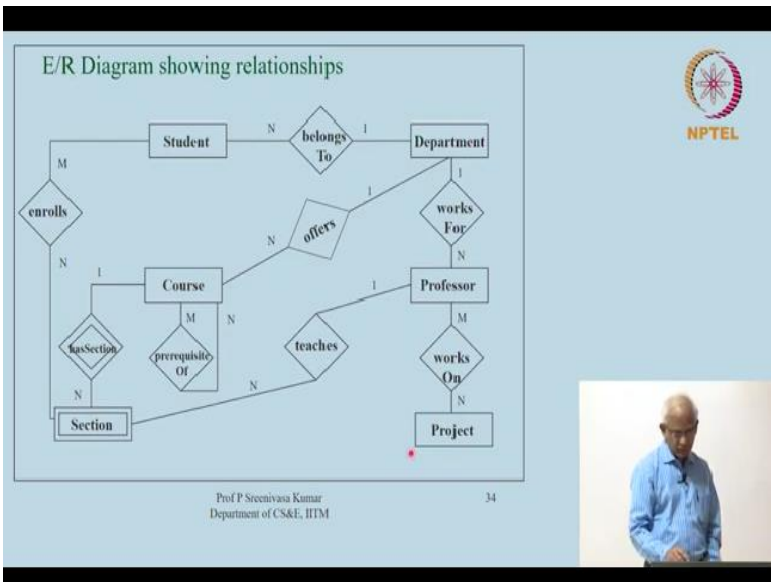
Then we have the department, department has a department number which serves as the key attribute and it is headed by a professor, so we call it, so we capture that way HOD attribute and then name of the department location and phone number. Course entity course ID, credits and name. To recall that courses are offered by department. So, we intend to capture that as a relationship because each of these things is a entity department entity exists and course entity exists. So later we will capture that the departments offer courses through a relationship.

(Refer Slide Time: 21:33)



And then professor a name or name Id, phone number, and maybe email address things like okay, I am trying to strictly follow what was there in given in the English text section is a weak entity, section ID, classroom and timings. Timing is multiple timing so, it captures basically as to at what times of the week the section needs. We have project which is having a sponsor, and an amount of money given and what is the start date and date and what is the idea of the project. Now, let us put all these things together into one diagram.

(Refer Slide Time: 22:25)



This is how that now only I want to remind you that because I am using slides and the slides will get cluttered if I write all the attributes around this, I have not written here, but normally what you do is you do not show this ER diagram like this, when you show the entity you also show all

the attributes around it. Likewise, if any relationships also have attributes, we have show all of them.

So just because the slide will get cluttered. I have shown these entities separately, and I am now showing the relationships alone. So your diagram showing relationships alone, but normally a residence will have all the attributes also written here okay, now let us focus on the various relationships that we have here. The student belongs to a department. So it is a many to one this one relationship, many students belong to one department.

And department offers several courses, so the one to many relationship and professors work for department. My relationship between professors and department works for and many professors work for a single department. And each professor was for exactly one department. So that is why it is one to many and then professors work on projects. So, this we have already discussed.

This is a typical many to many relationship because professors work on many projects and each project can have several professors associated with that. So it is a natural many to many relationship. So that is why it is shown as m and n and professors okay. So now courses have sections. So, courses are not directly you know taught is the sections that are taught. So section we have introduced as a weak entity earlier.

So that section has section ID classroom and timing right. So and it is a weak entity owned by course. And this is the identifying relationship. So professors teach sections and students enroll into sections okay. So there is an enrollment relationship, which is a many to many relationship because many students enroll for into one section. And a student typically enrolls for many sections, and the professor teaches more than one course of a section so.

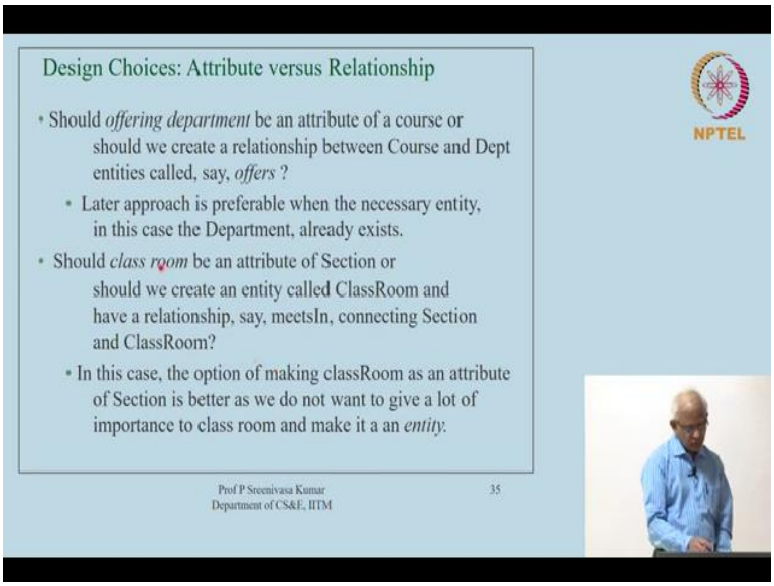
So it is one too many. So, all these are relationships and perquisites of is a relationship that exists between courses. So, we should ideally right the role name here, I have not written the role name course participants as a course and as a prerequisite. So participates as in what role does it participate as a participants as a prerequisite, participate as a as a course. So that is the prerequisite of relationship, it is we call this as recursively relationship in database pallets okay.

So, I hope it is now clear to you as to how you can construct this ER diagrams and then take them for discussions with the stakeholders, so that you can understand and the entire requirements and we can also convey to them as to what we have understood about the domain, what we are going to represent in the domain later on, they should not ask for something which we have not discussed, right and then finally.

Because all these design decisions are best made, data model requirements are best made at the design stages and later on if they start asking for additional things, it becomes a little bit messy to handle the database system in creating the database system okay. Now let me take you through in a tutorial what we will do is to. So, I want you to have you know develop the skill of reading ER diagram and understanding that okay.

And then write down in plain text about a particular domain and translate that into an ER model or given any you know, as situational description, translate that into an ER model. So, these 2 things are what you should aim for. So, I will devise a few exercise like that and then we will have a tutorial session sometime next week okay.

(Refer Slide Time: 27:35)



So, let us come to a few questions, which we will encounter as we are trying to develop this designs. One of them is attribute versus relationship. To give a concrete example this dilemma

between attribute versus relationship, should offering the department be an attribute of a course or should we create a relationship between course and department entities and call it offers, this is what actually we have done in the previous figure.

But why should it be like that, why cannot we simply you know, create an attribute for the course and call it offering department and it will have a value right. So, then it indicates what is the offering department. So, we will be done with that, why should we go for a relationship. Now, whenever the other entity see for example, so, what we are now talking about is for a particular entity like course, should some piece of information be modeled as an attribute or as a relationship.

So, this is you know again this is a design choice, if the if that information is actually you know, corresponding to a particular entity that already exists then the approach of creating a relationship is to be chosen. And if it does not then you can go within attribute. So, that is the kind of choice that we have. So, it is preferable if the for example, in this case department exists as in other independent entity.

So, in this case, it makes sense for us to create that relationship like offers and then put it rather than making offering department as an attribute of course, instead we can actually show that the association between course and department through a relationship okay. If it is some other information, which cannot, you know, naturally get linked up with any of these existing entities, then you choose the option of modeling it as an attribute basically.

Now, another kind of choice should classroom be at is related situation is should classroom be an attribute of a section or should we create an entity called classrooms and then you know make a relationship saying needs in and then connect them to, this also we can do actually, we can create an entity called classroom which is an entity set consisting of all classrooms and then say that this particular section meets in a particular classroom.

So, you can create a meets in relationship and then connect these 2 entities. That option exists, what we have done in this case, if you go back to the picture is simply we made a classroom as

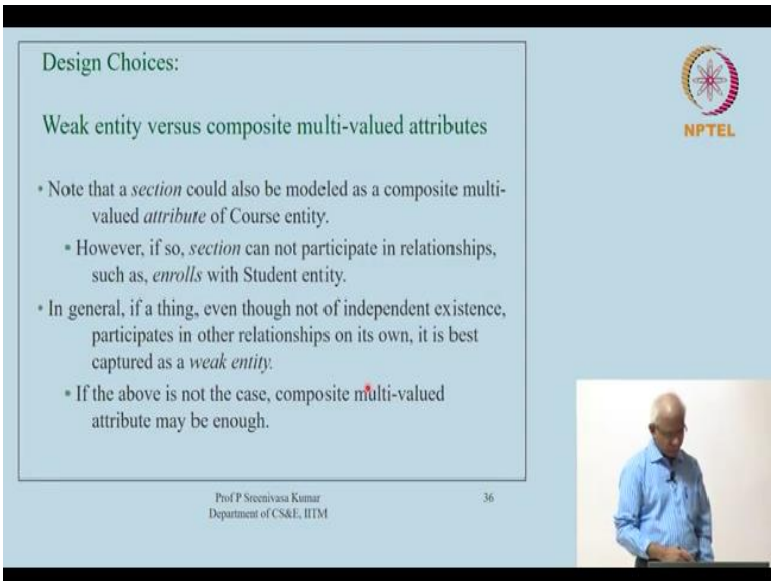
an attribute. We did not model classrooms as entities, we simply captured that information as an attribute of section. Again, this is a design choice. And in our information system, we are not keen on you know, keeping track of lot of other information about classroom as to, you know, what is the equipment that is there in that particular classroom.

Who is in charge you know for upkeep of the classroom, and know who has the keys where the keys have been issued. Let us say we have our information system actually is supposed to keep track of that information also, if that is the case, then it kind of makes sense to model the classroom as an entity because we have a lot more other information to keep track of and once that classroom becomes an entity, we can then create a meets in kind of relationship and connect up this section with the classroom entity.

But since we do not want to, but in this particular you know specifications that we have, we are not keeping track of a lot of other information about classroom and so we have chosen the route of modeling it just as an attribute okay. So these are the some of the design choices that we will have to make. When we construct the model, it all again depends on the domain knowledge that we are supposed to capture okay and the usage of that information.

So, in this case, the option of making classroom as an attribute is better as we do not want to give a lot of importance to the classroom and kind of make it as an entity because we do not have additional we are not required to keep track of additional information about classroom okay as per our specifications okay. So, this is one interesting point.

(Refer Slide Time: 33:19)



Then let us look at this other situation, weak entity versus composite multivalued attributes remember that we have attributes and multivalued attributes, multivalued attributes or attributes that can take multiple values right a set of values and then composite that means they have components in that. So, for example, a section actually could also be modeled as a composite multivalued attribute of course.

Do you realize that I can model a section multiple sections. So basically, it is multivalued. And since while I am keeping track of some information about section I will make it composite. So using a composite multivalued attribute called section, I can attach it to the course, and then make this action as a composite multivalued attribute of course, is that a good design choice okay.

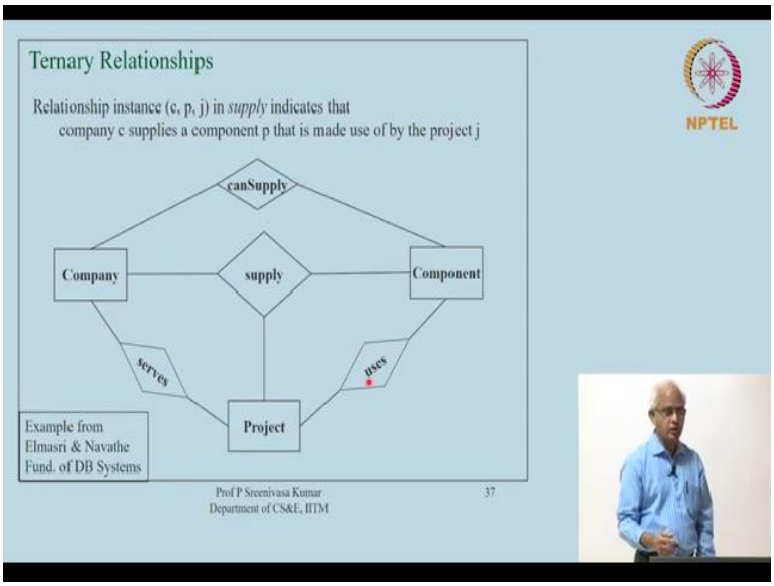
If you do that then they call that attributes do not participate in relationships. So, this section, you know, becomes a multivalued attribute the need cannot participate in so what about enrolls and then what about teacher and teaching that relationship and all that it becomes difficult to model those things now. So, it all again depends. So, if that particular entity is that information is required for other things like relationships and all that it is best model as a weak entity.

Otherwise you could even go ahead with modeling it as a composite multivalued attribute okay. For example email address could be model as a composite multivalued attribute multivalued

because it has user ID and domain name and multivalued because it has typically person has multiple email address okay, I mean just you could also think about some other examples like that. So, in general, if a thing even though not of independent existence part is debates and other relationships on its own like our section late it is best captured as a weak entity okay.

If this is not the case then you can go ahead and moderate as a composite multivalued attribute. So, it all depends on the domain which we are modeling and the usage of this information in the domain and based on that we are to make all this design choices okay. Now, finally, to kind of wrap up this module, we have will spend a small amount of time on what are called temporary relationships.

(Refer Slide Time: 36:46)



Often the kind of relationships that we encounter in modeling or frequently binary kind of relationship. But occasionally we do you know have to make use of n-ary relationship. So, here is an example of a ternary relationship this example is from Elmasri and Navathe fundamental database systems book. So basically what we are seeing here is that there is a relationship called supply okay.

And that has 3 entities participating in it the company, project and a component and we have also thrown in the corresponding binary there are certain binary relationships also we can bring into picture like okay like when example okay, so a company can supply a particular component. So

we want to capture that binary relationship and a project is some activity and a project uses a component.

If the component is used in the project then we have this binary relationship called project users component and if a company is serving is you know supplying some component to a particular project then we call that as company serves the project okay. So, that is a binary relationship between company and project, if it is supplying some component to the project, then we call it company service the project.

So, these are the familiar binary relationships, here is a supply relationship where the instance c p j in supply basically indicates that the company c supplies a component p that is made use of by a project j okay, it is the all the 3 things are associated together is what is captured by the supply okay. So, if you want to capture this particular situation we need a ternary relationship. Now, if a tuple like this c p j exists in the supply, ternary relationship.

Then in some sense it should be obvious to you saying that this c p you know could exist and can supply c supplies component p c p would exist in sub can supply and then you know p j could exist in sorry j p j p would exist in project users the component, j is the project, p is the component. So, j p exists incompetent and in a similar way c p c p exists in company sorry c j c j could exist in source.

So, the corresponding entities like this. So, the presence of c p j would imply the presence of certain things here in the binary relationships, but the other way around may not always be true. So, supposing we have c p in can supply and j p in users and c j in source may not together actually imply that c p j is in supply okay. So, this kind of illustrates that the having the 3 binary relationships is not a replacement for having the ternary relationship.

You sometimes need to have the some ternary relationships because it see the moment you just say let us say c p j p and c j exists, then that does not really imply that the company c supplies component p and j uses that particular component, it does not imply that think about it. So, the binary relationships together and do not convey the same meaning as supply. And so, when we

are dealing ternary relationships ternary and in general n-ary relationships will have to pay attention to how what are the domain, you know details.

And then appropriately capture the we may want to, so, capture the make a design choice whether the binary relationships are enough or you necessarily have to go for the n-ary relationship okay, so there are some more interesting considerations to be thought through when we are using n-ary relationships okay.