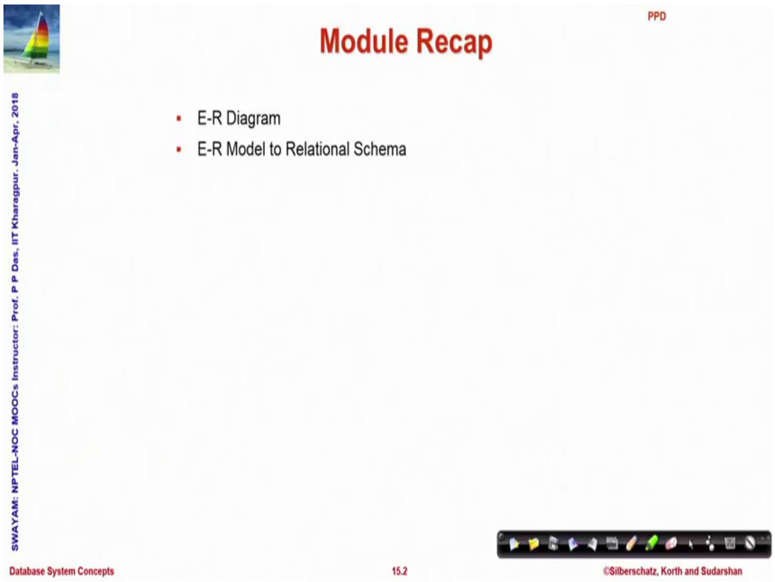
Database Management System
Prof. Partha Pratim Das
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 15
Entity-Relationship Model/3

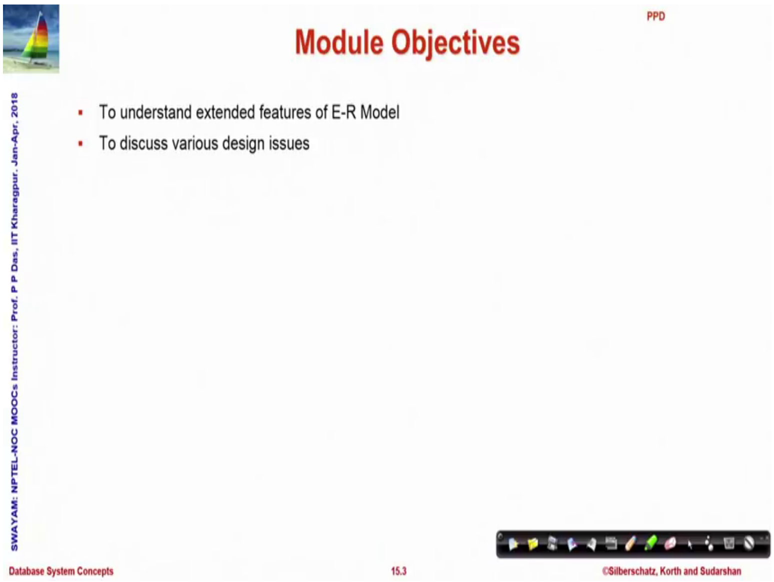
Welcome to module 15 of Database Management Systems. We have been discussing about entity relationship model and this is the third and the closing module on this topic.

(Refer Slide Time: 00:37)



So, in the last module we have discussed about E-R diagram and we have also seen how E-R model can be converted to a relational schema.

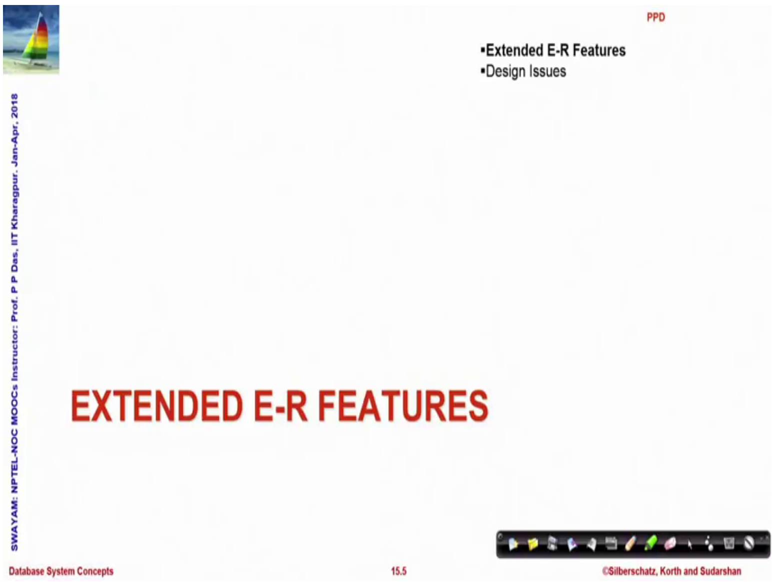
(Refer Slide Time: 00:47)



In this module we will try to go through a few extended features of E-R model, try to show some of the more complicated situations how they can be modeled in the E-R model and along with that we will discuss a variety of design issues that will follow.

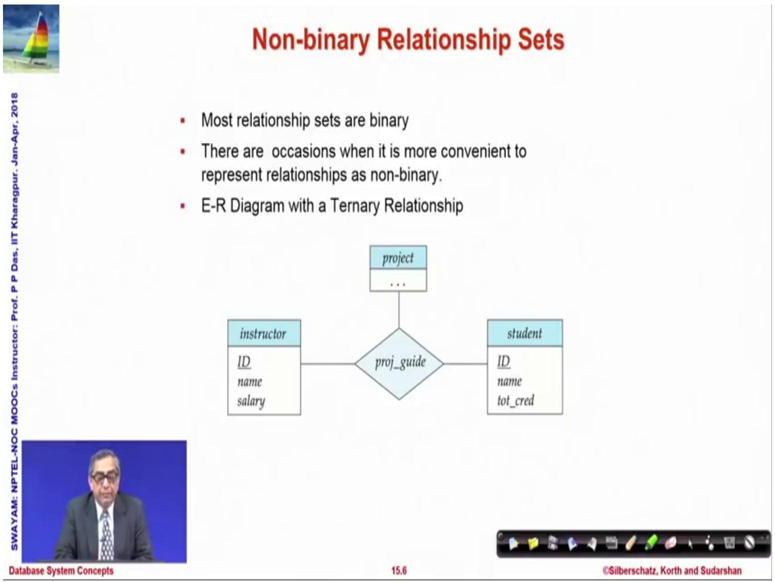
So, these are the outline.

(Refer Slide Time: 01:19)



So, we start with extended en tity relationship features.

(Refer Slide Time: 01:23)

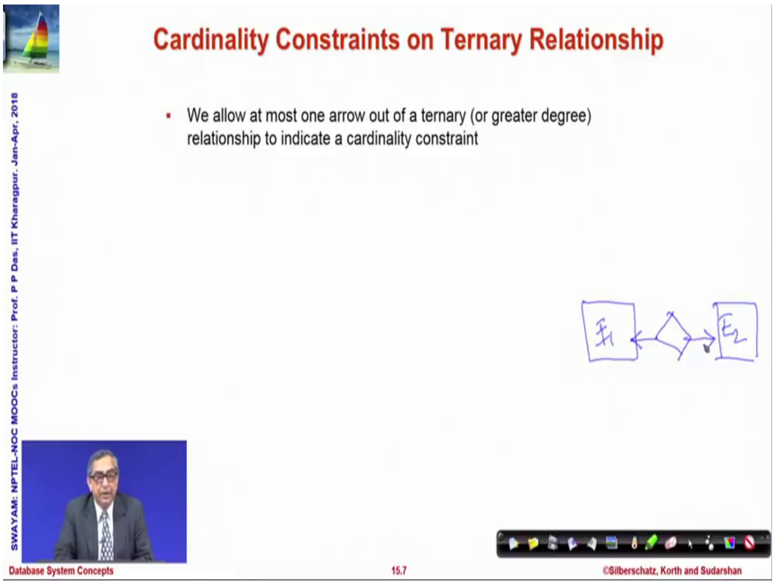


The first that we note is so, far in the entity relationship model we have talked about relationships between two entity sets. So, we have talked about the student attending courses or instructors advising students and so, on. So, such relationships are naturally called binary, but it is possible that more than two entity sets, let us say three entity sets could be involved in the same relation and we show an example here where we have three entity sets instructor, student and project.

So, the project entity set is a list of projects being done by the students or to be done by the students. So, the relationship project guide is a relationship between the guide who is an instructor, the student who will do the project and the project that has to be performed. So, all three together define this relationship; so, in such cases it is possible in E-R model that we can represent it conveniently as a non binary relationship.

Now this is an E-R diagram this is called a ternary relationship R.

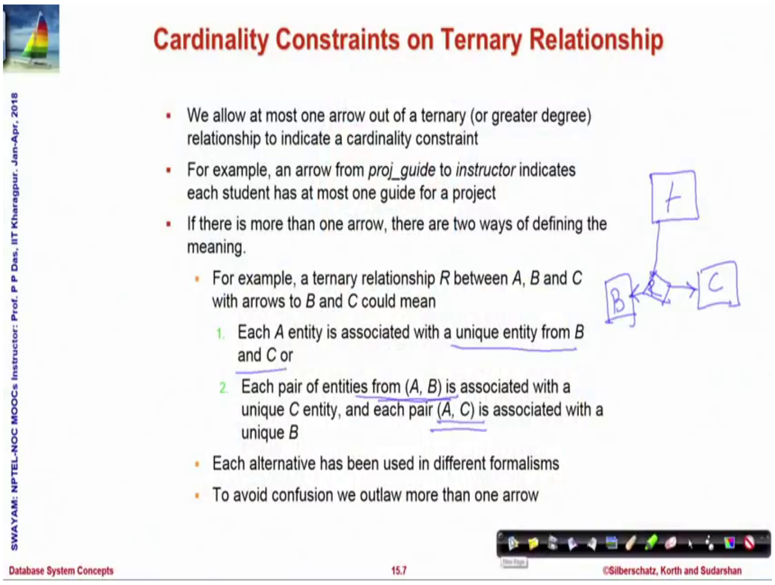
(Refer Slide Time: 03:02)



Now we have talked about cardinality constraints on binary relationship one to one, many to one one to many and many to many. And we specified that if we have a binary relationship say this is one entity set and this is another entity set and we have a relationship. So, if we just connect them it means a many to many relation, but if we have an arrow on one side entity set then it means on the arrow side its one.

So, this is from entity set E 1 to E 2 this is one to many; we could have arrow at both ends and; that means, one to one. Now the question is how will that work out what will be the meaning of arrow in terms of a ternary relationship.

(Refer Slide Time: 03:55)

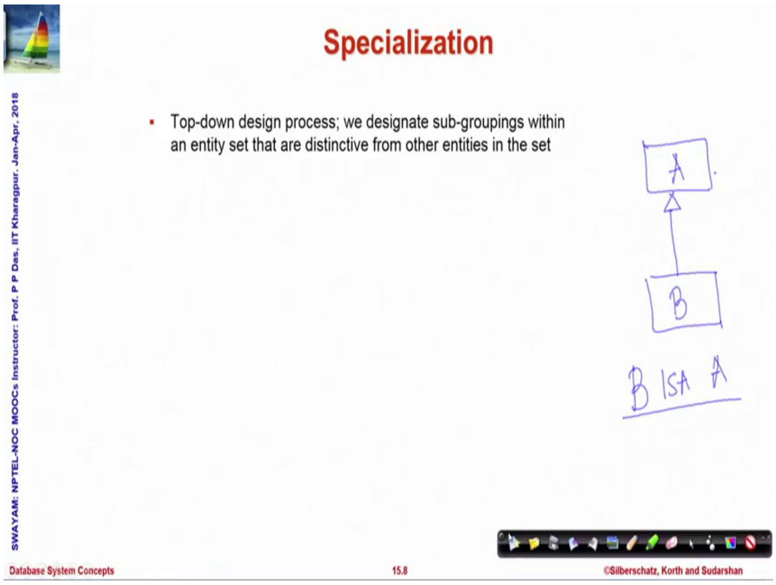


Now, in the case of ternary relationship or this would generalize to relationships of higher degree whether where more than three entity sets may be involved. We put a restriction that we will allow at most one arrow out of the ternary relationship. So, we could have only if we if we look at if we look at say this relationship then we could have an arrow only at this end.

But multiple arrows are not allowed and the reason is certainly to keep the semantics of the cardinality meaningful. For example, if we have a ternary relationship between A B and C let us say this is a A this is B and this is C and we have a ternary relationship between them. And let us say if we have a more than one arrow; there for example, suppose then we have say an arrow to B and an arrow to C the question is how should we interpret that?

Should we interpret that an entity of entity set A is associated with unique entity from B and C together or should we associate, should we interpret that the entities formed by the pair a B and the entity formed by the pair A C are uniquely related. So, there is multiplicity of interpretation; if we allow more than one arrow in case of eternity or higher degree relationship. So, what will follow for simplicity in this course and that is what is followed often in practice; is in case of a ternary or higher order relationship only one arrow will be allowed in that in it in that relationship.

(Refer Slide Time: 06:20)



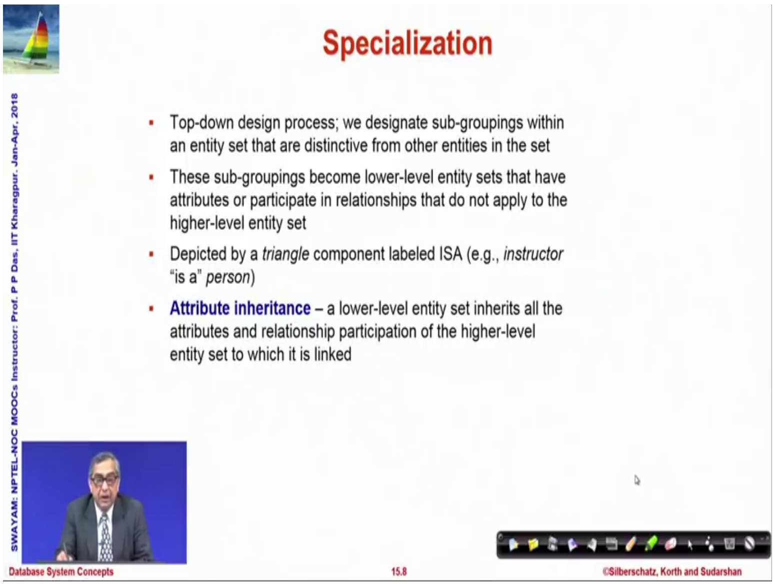
Now, let us talk about specialization those of you who have some background of object oriented systems would be familiar with the notion of specialization and generalization in object oriented system. So, we say that if we have a certain concept say we have a concept called person and then we say that a student is a person; what we mean is a student is a specialization of person. And person is a generalization of student and in that process student inherits all the attributes of person, but in addition the student may have some specialized attributes.

So, what it means that if you look from the perspective of such specialization; say if I draw like this; this is an entity set A and this entity set B and to mark the specialization we show an arrowhead with a blank triangle at that end. So, if we by this what we mean is B is a A. So, B inherits all the properties of A, but can have some more properties. So, if you look at all the entities that A may have you will find that a subgroup of the entities in the entity set A have some additional common properties.

So, if A is set of persons and B is a set of students then A may have entities who which represent people who are not students; who are employees, who could be retired and so, on, but there will be a number of entities. So, who have the commonality of being student they are enrolled in certain course of certain university and so on.

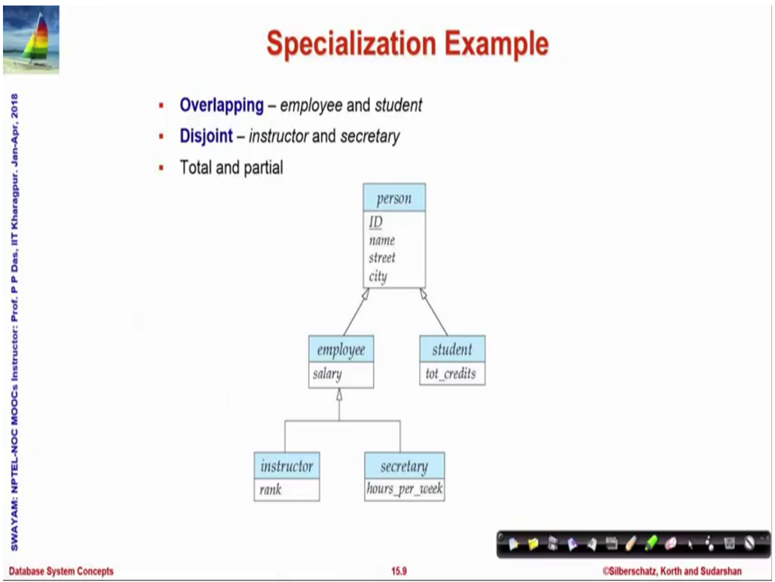
So, in terms of the E-R diagram E-R model what we do is we try to look at the entity A and move top down. So, that whenever we find a group of entities which have certain commonality; we move them into a lower separate, specialized entity set and relate these two entity sets to the specialization relation.

(Refer Slide Time: 08:54)



So, these sub groups form lower level entity sets and as I said that it is designated in a certain way. And as in the object oriented system the lower level entity set inherits all the attributes and relationships of participation of the higher level entity set. So, here is an example.

(Refer Slide Time: 09:16)



Can say the person at the so, called root of this hierarchy of specialization; it has a set of properties ID, name, street, city and employee is another relationship; another entity set which is a specialization of person relation person entity set.

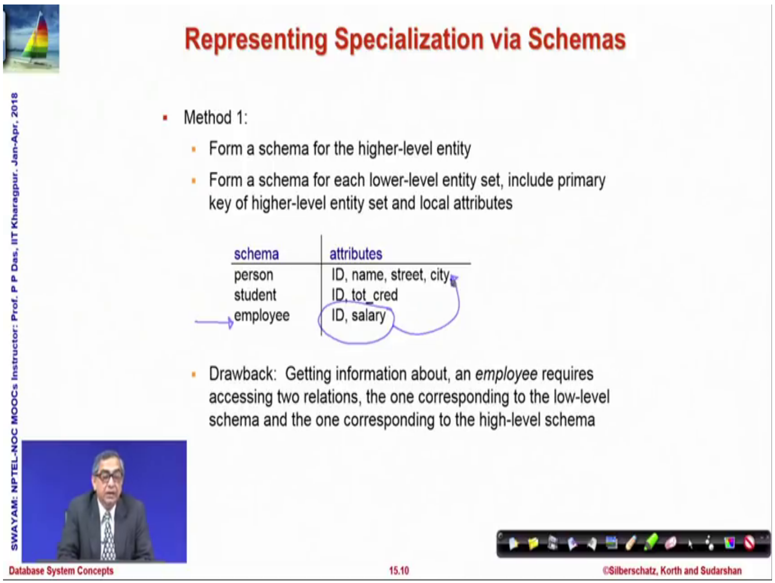
So, employee inherits all attributes ID name street and city, but in addition the commonality of the entities in the employee entity set is a fact that all of them have a salary attribute. A similar entity set student is a specialization of person where the again all attributes are inherited, but there is a common attribute called total credits which is common for all the students, but is not available or common for the persons in general.

And as you can see that it could be hierarchical it could go further down employee could be specialized into instructor and secretary. Again by the rule of specialization instructor will inherit all attributes of employee which means that it will inherit attributes of person; which imply has inherited plus the employee specific attributes salary. So, it will inherit all five of those attributes and then it adds another attribute which is specific for the instructor which says a rank which is another specific attribute that you have.

Now, when we specialize a certain entity set into two or more entity sets like we specialized person in employee and student; then there could be different situations that could exist for example, certain entity may be a member of both employee as well as student. If that happens then we say that their overlapping entity sets or they could be disjoint where no member of instructor would be a member of the secretary and vice versa.

So, we say that this disjointness tell us that no instructor can be a secretary and no secretary can be an instructor. Whereas, overlapping specialized sets denote that well an employee may or may not be a student, but it is possible that some employee is also a student and vice versa and that is how we will represent this. And we will see that when we specialized the specialized entity could be total or they could be partial.

(Refer Slide Time: 12:06)



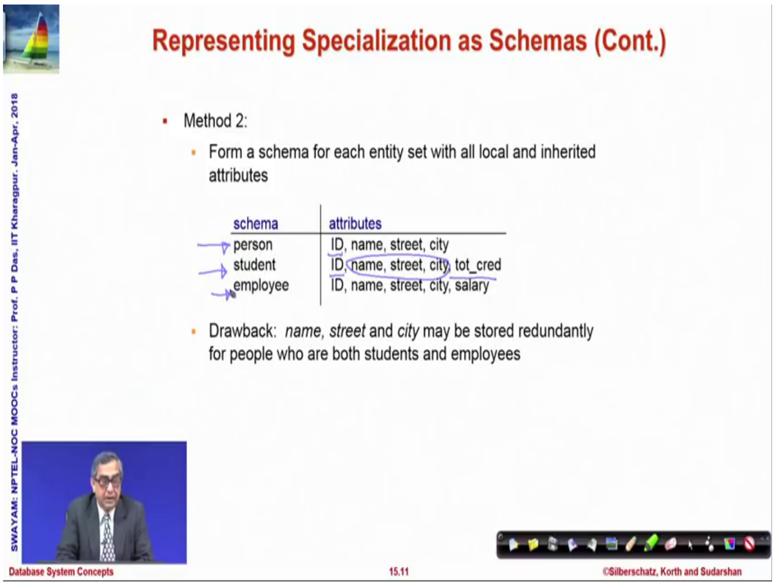
We will talk about that totality and partiality little later; let us look at how do we represent this information in the relational schema because as we have seen that whenever we have a E-R diagram; it is important to find out what is the relational schema that will be corresponding to that E-R diagram or E-R model. So, here we could do this in two ways one that we are showing here is form a schema for the higher level entity. So, form a schema for the person as you can see here that person is described in terms of four attributes and this form a schema for each of the lower level entity set where you include the primary key of the higher level entity set.

So, when you are forming the schema of person of student which is a specialization of person you include the ID which is the key of the higher level entity set person. And along with that you include the so, called local attributes or attributes which are specific to this low level entity set in this case total credit similar thing happens with employee.

Now this representation is in a way optimized because you are representing the information only once when it is needed, but the drawback is if you have to find out information about say employee; then you will not only have to access the employee entity set or the corresponding relation in the relational schema, but you will also have to access the parent or higher level entity set to get the attribute values which are inherited. And if you have a multi level hierarchy as we have shown this could involve accessing multiple relations to find information about a single entity in an entity set.

So, this is an in terms of data representation this is an optimized representation, but it has the overhead of having to access multiple entity sets to get information about certain entities.

(Refer Slide Time: 14:22)

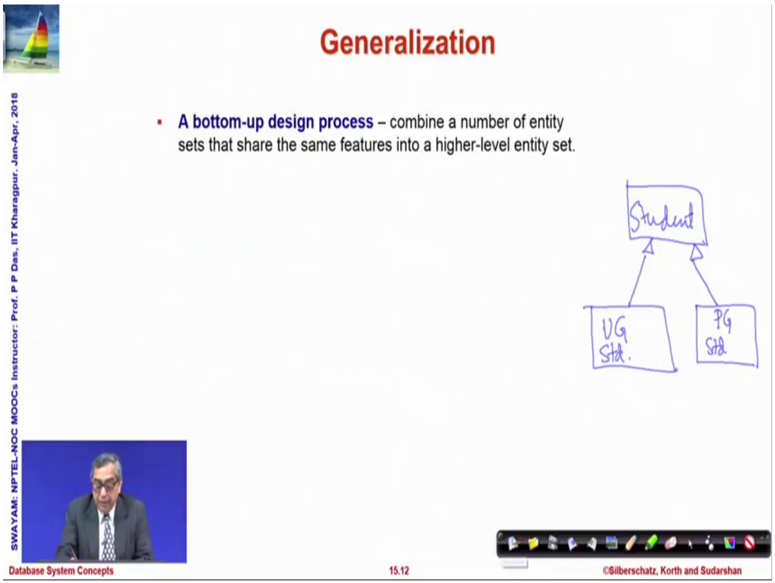


An alternate scheme would be that based on the hierarchy of specialization, you assume all attributes as they are inherited and then represent every entity set in full. So, when you the representation of person does not change, but when you represent student now earlier you are just having ID and total credit; now in you include all entities that are inherited.

Similarly, you do the same thing; so, you have the all entities of the parent to all attributes of the parent entity set as well as the local attribute of that specific entity set. Now this naturally makes it easy to extract information from a for a single entity set, but at the same time, you are storing the same data redundantly for people who are having overlapped representation. So, if we have as we know student and employee overlapped. So, the same entity will happen in student as well as in employee; so, it will the information of the common attributes name street city etcetera they will occur in both these tables in the design.

So, these are two methods and now we have just given you the relative advantages and its advantages of the same and based on a particular situation you will have to choose what is a good method to represent.

(Refer Slide Time: 15:51)

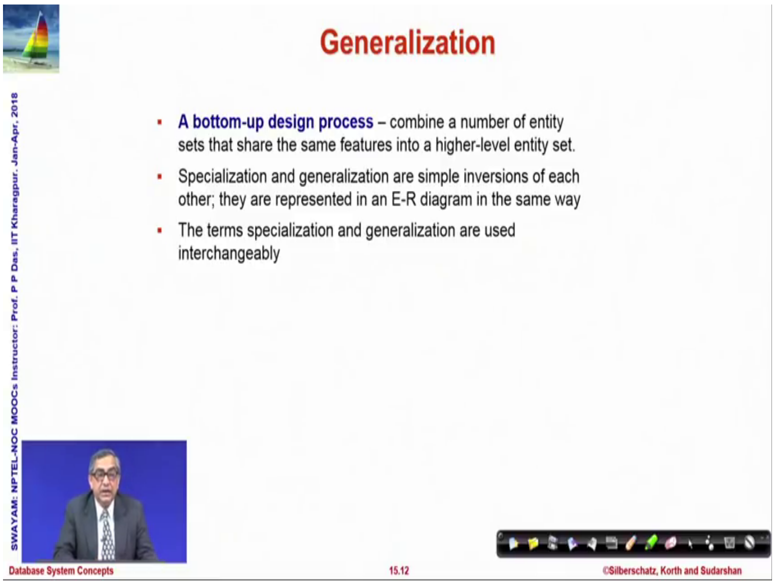


You can look at now you know from the object based system that if you have a specialization hierarchy you can think of it as a generalization hierarchy also.

The generalization hierarchy goes in a bottom up manner. So, instead of actually starting with an entity set and finding out subsets of entities which have greater commonality between them and putting them as specialized, you could actually group them in terms of finding out what they share and create a higher level entities. For example, the way I am saying is let us say that I have one entity set which say UG student and have another entity set in the same university which say PG student.

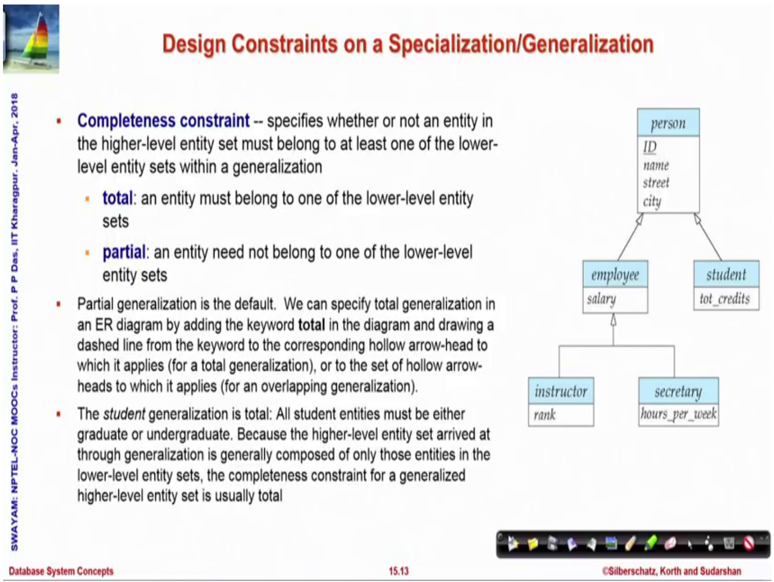
So, there are both of them are students naturally they are disjoint a person cannot be UG as well as PG student at the same time. And once you represent that you find that well there are a lot of information which are common between these two entity sets like the student roll number, name and so on so, forth. So, you could choose that well you instead of having them as two separate entity sets, you could extract out the common attributes and put them at a higher level entity. So, all that you are doing is instead of going top down you are going bottom up in the whole approach.

(Refer Slide Time: 17:21)



So, if you do that then there you can easily see that specialization and generalization is are inverse of each other and they are used interchangeably in terms of the relational entity relationship design.

(Refer Slide Time: 17:35)



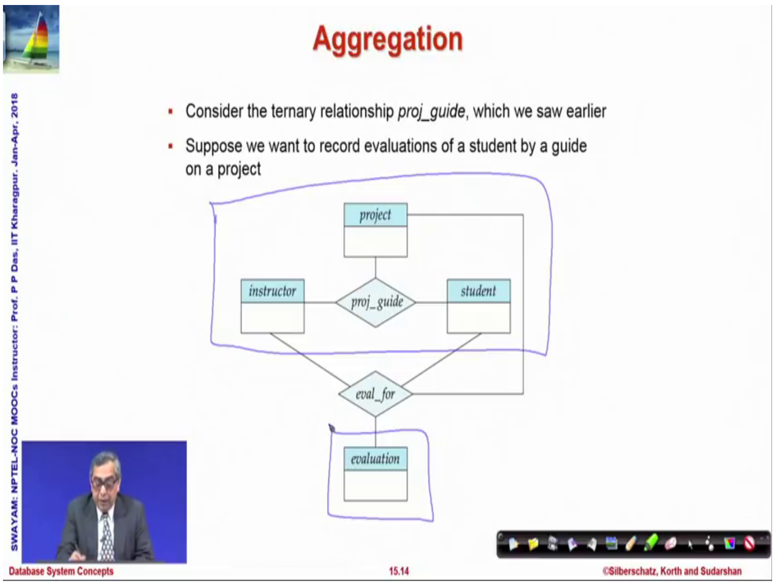
The other constraint that you can identify, you should identify is the constraint of completeness which say that if I have a entity set say a person. And then we have specializations of employee and student; the question is for a higher level entity set is it necessarily that every entity will be represented in one of the or more than one of the lower level entity sets.

If that is guaranteed that an entity must belong to one of the lower level entity set we say that it is a complete specialization. But if it is that a high level entity is may or may not be featuring in a entity set, which is at a lower level then we will say it is a partially specialized hierarchy. So, the both of these are possible depending on different situation that we have. So, by default we assume partial specialization and. So, if we want to say a certain specialization is total we will have to write the keyword total by the side of the arrow head that is representing the specialization hierarchy.

You can say that the example I talked of in uniting unite undergraduate and graduate or postgraduate students into the entity set of students, gives you a hierarchy which is total because every entity in the entity set student must be either a UG student or a PG student; it is not possible that I have a student who is neither a UG student nor a PG student. So, every high entity at the higher level entity set student must feature in one of these two specializations.

So, therefore they are necessarily total in the relationship. So, this is the completeness constraint that you can think of.

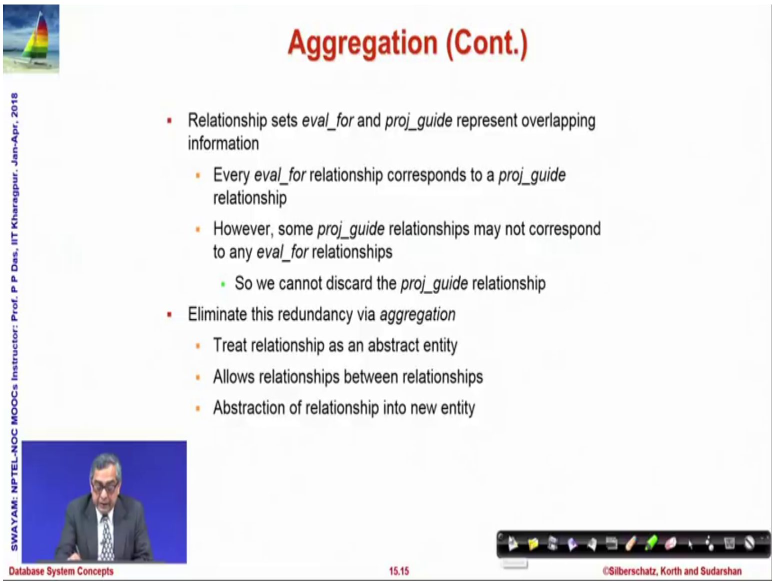
(Refer Slide Time: 19:52)



Moving on let us talk about another feature which is known as aggregation; the situation is like this we have already talked about this part of the diagram which is a ternary relationship which relates project instructor and student.

Now, let us say once the project progresses you would need to add evaluation to that. So, there is another entity set which represents evaluation and how we are grading or putting marks and so, on. So, naturally the evaluation of a student will be dependent on the project, the student and the supervisor and that will relate to the evaluation. So, evaluation eval for the relationship is necessarily a relationship between four entities or four entity sets so, to say.

(Refer Slide Time: 20:47)



Now, the question is how do we represent this information? The relationship sets eval for project guide the two that we saw if we just want to recall once more. The project guide involves three of the relation entity sets and the eval for relates to four of the entity sets. Now every eval for relationship corresponds to a project guide relationship; that is if I have an entity in eval for relationship, I will have a corresponding entity in the project guide relationship which specifies the student project and the instructor.

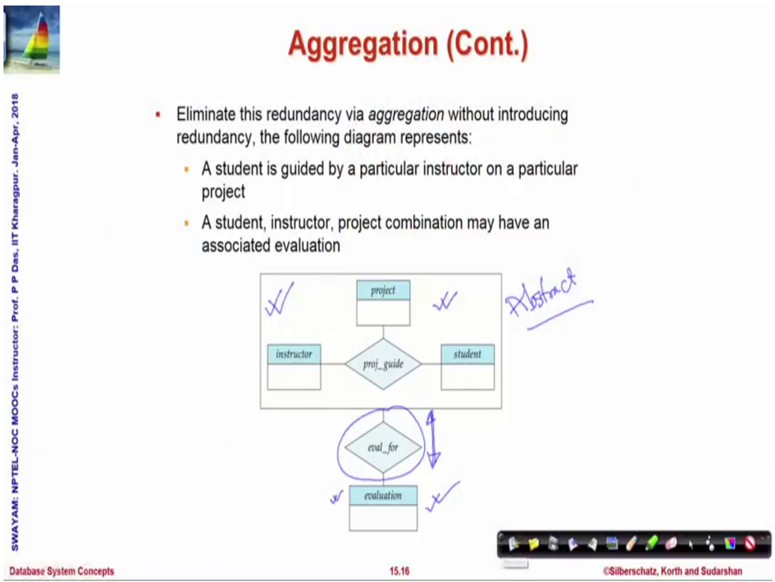
So, but it is other way it is possible that some project guide relationship may not correspond to any eval relationship; that is it is possible that there is a allotted project by a student with a particular instructor which has yet not been evaluated; the evaluation process is not complete or the time has not come.

So, if we have to represent the information only for the eval for relationship; we will get partial information because it is possible that some entities in eval for does not have the eval for information do not feature there, but need to be preserved because I need to remember the project guide, instructor, the student and the project. So, we need to keep both duplicating the information.

So, we can use aggregation to eliminate this duplication of information or redundancy of information. So, what we do is we treat the first relationship; the project get relationship as if it is an abstract entity and then you allow relationship between two relationships; this is something we did not do before relationship so, far has always been between entity sets.

So, what you can see that project guide relationship itself as if it is an virtual entity; it is an abstract entity and then you allow the relationship between the project guide and the eval for relationship which relates to the evaluation entity set this really this shows I mean I will just show you in the diagram.

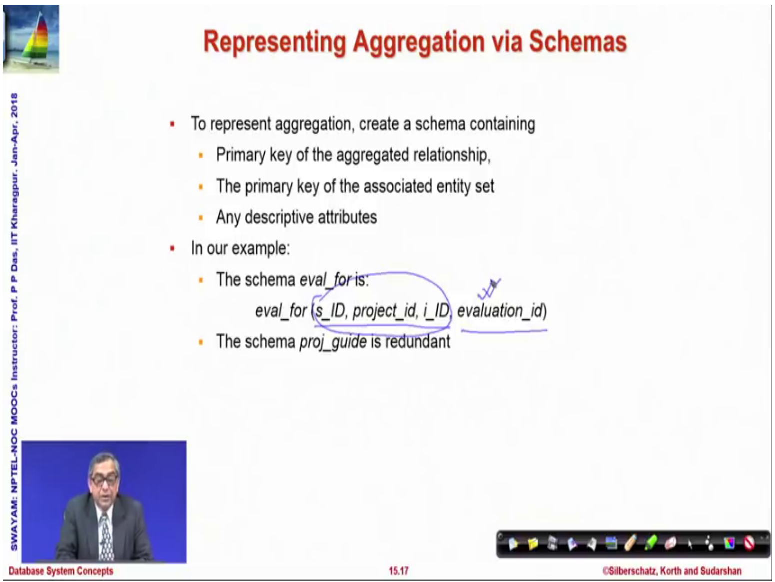
(Refer Slide Time: 23:08)



So, this is how it will now work out to be. So, this is the abstract project guide entity set which is an abstract entity set because it is actually relationship. And that relates to eval for which on the other site has the evaluation. So, what will happen is a student is guided by a particular instructor in a particular project will feature in this abstract entity set; which relates the three entity sets project student and instructor. And if it has an evaluation then the; this through this relationship it will be represented and the evaluation value will exist.

So, we know that if a project is evaluated then it certainly have a corresponding entity in the abstract entity set project guide, but the reverse may not be true I may have an entity in the project guide entity set which does not have an evaluation. So, by using this aggregation model; I can represent the information of this situation model this situation more accurately than I could do otherwise.

(Refer Slide Time: 24:30)

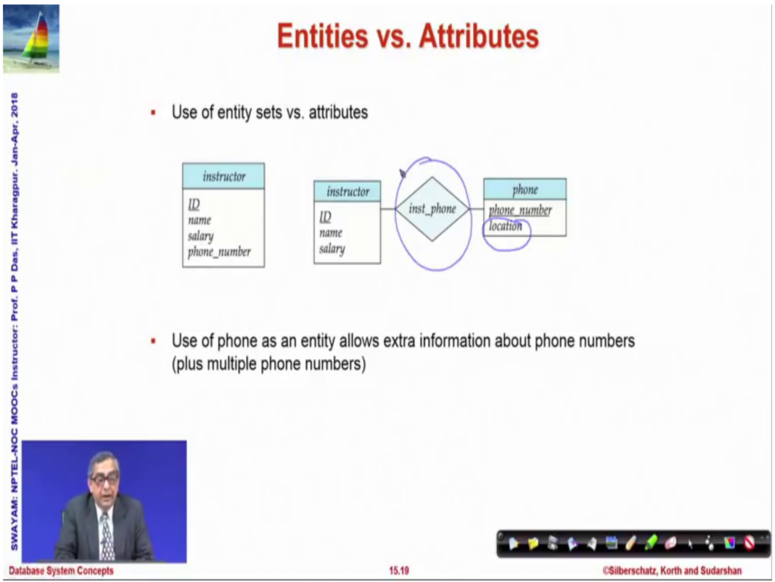


So, this can be represented again how to represent this in terms of the schema? So, what we do we represent the aggregation we create a schema containing the primary key of the aggregated relationship. The primary key of the associated entity set and all the other descriptive attributes and put them together.

So, in our example the schema would be eval four and that schema will have these are entities these are attributes of the aggregated or abstract entity set which is coming from the student project and the instructor entities and this is for the evaluation ID. So, we put this together; so, now, you can see that all of these are related representing who is the guide of which student in what project and if this exists then this gives you the evaluation.

So, naturally once this has been represented; the project guide schema by itself becomes redundant and therefore, it can be removed. So, this is a process through which we come to the decision of actually having this schema to represent all the required information. Naturally if the evaluation is not done then the evaluation ID for in eval four will not exist and that will be a null showing that it is not present right now ok.

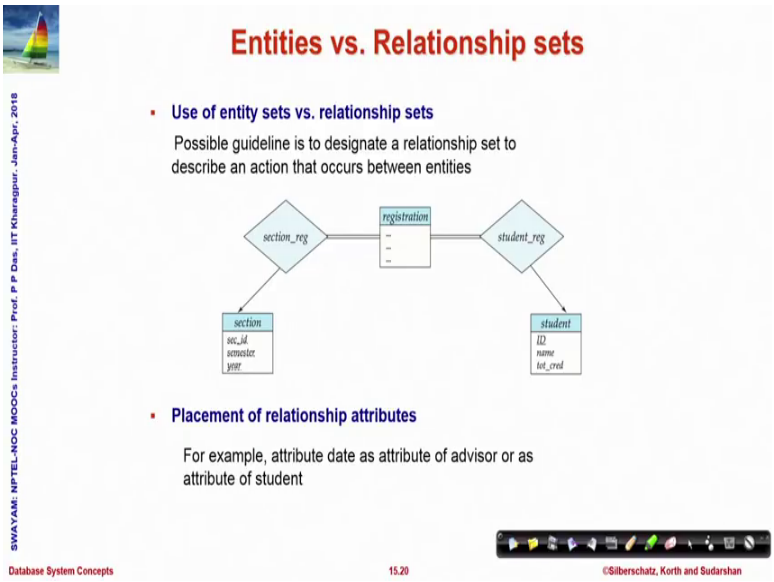
(Refer Slide Time: 26:25)



Now, given these basic features as well as the extended features let me talk about a few design issues which will be required to see what kind of information that that the different challenges that we have seen so, far. For example, we have seen the case of multivalued attributes.

So, and the way we can represent that is using that multivalued attribute as a separate entity set like the phone number which also has the advantage of having its own added information. For example, once we do this then not only I can have against the same instructor ID; I can have multiple phone numbers, but I can have location for each one of these phone numbers. And I make use of this relation relationship that I create which allow me to represent this multi valued attribute.

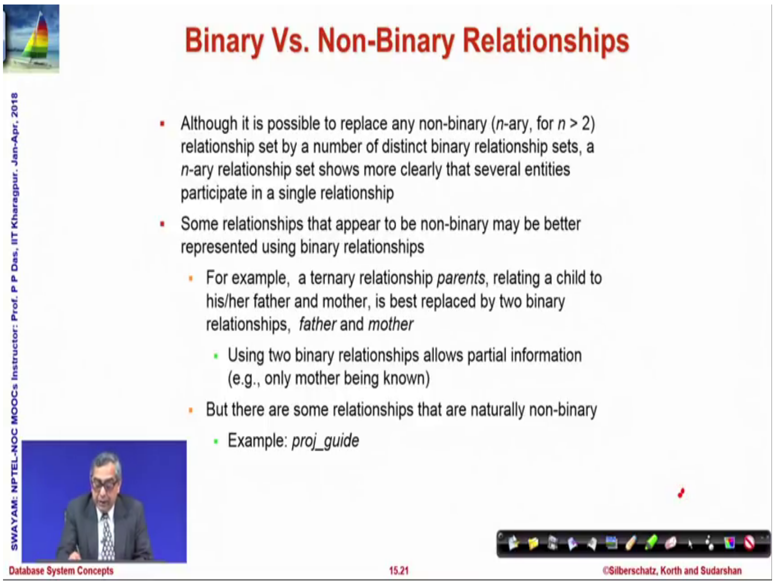
(Refer Slide Time: 27:30)



So, this is a common technique that will be used frequently in such cases you can have entities versus relationship for example, if we have inform we need to keep information about registration how students register to different sections; then we could represent registration as an entity set and have different relationships of section registration which specify how registration is related to section and student reg; which specify how do the station is related to students to represent that kind of information.

We can have placement of relationship attributes also attribute date we have talked about as an attribute of adviser to designate as when that particular instructor became adviser of a student is a common situation that we have already seen.

(Refer Slide Time: 28:25)

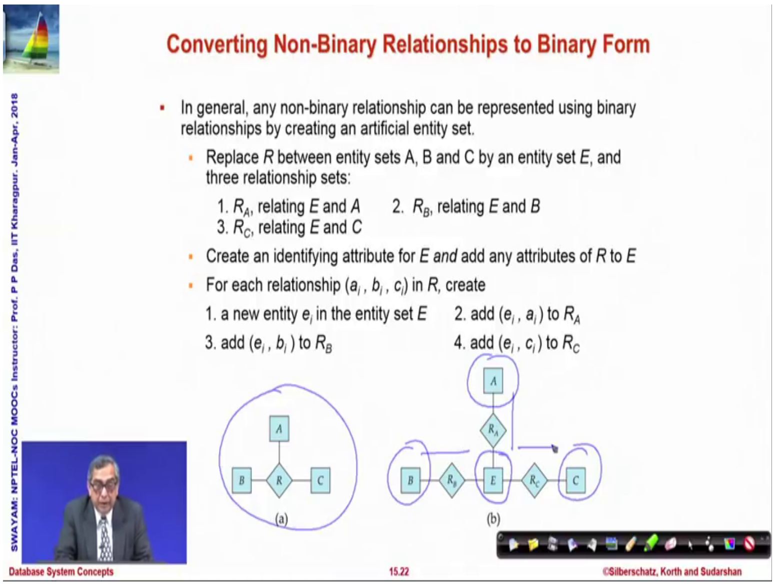


There is also question of the choice being made between the binary and non binary relationship ternary or higher degree. Now as it turns out that it is possible that you could represent ternary relationships directly or you can decompose that. For example, a ternary relationship can be decomposed in terms of two binary relationship; for example, let us say if we talk about persons then person every person has parents; so, he or she has a father and a mother.

Now, if we represent this as a ternary relationship then the one difficulty that we have that a person must have both the father and mother to be represented there. For example, if we can come to a situation where only the mother is known, the father is not known I will not be able to represent that because it will always have to come as a triplet of three persons thel the person under consideration her father and her mother, but if I represent the person and the father in one relationship; the person and the mother in another relationship, then I can take care of the situation where when one of the parents are known; I can still represent this.

So, there are certain tradeoffs which can be done between the choice of binary and non binary relationships, but; obviously, there are certain relationships which are inherently non binary for example, the project guide example we have seen. The project guide information cannot be decomposed without certain loss of information to be represented by say the instructor and the project and another relationship between the student and the project it really that does not represent the same information.

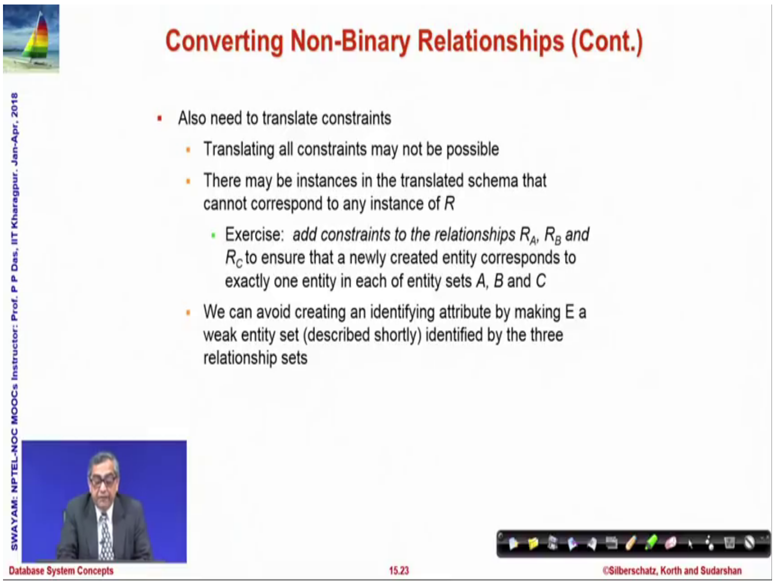
(Refer Slide Time: 30:20)



So, in general you can convert a non binary relationship by in the binary form by doing this. So, this is a ternary relationship being shown and for doing that these are the three entity sets involving the ternary relationship and to make decomposite into a ternary relationship; what we do is into binary relationships we inject a new entity artificial entity set E and then we define three different relations between them.

So, which individually relates to the entity sets A B and c. So, this is a standard decomposition and you can easily understand that A B and C in our earlier example could all be persons and R A could mean that father of R B could mean mother of and so, on. So, I can do it in decomposite in this manner and represent that.

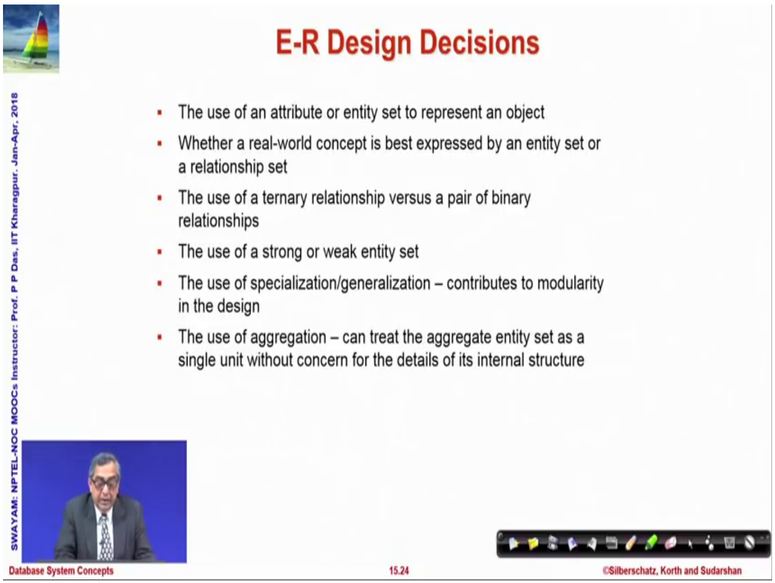
(Refer Slide Time: 31:24)



Now while we do this decomposition we will also have to remember in that; we need to translate all constraints that are present for the ternary relationship. And often times it may become difficult to translate all constraints, it may not be possible and there may be instances in the translated schema that cannot correspond to an instance of the original relationship.

So, we will have to avoid we can we will have to take care of this situation by identifying attributes. And making use of the weak entity sets which we have already seen in our earlier discussions.

(Refer Slide Time: 32:09)



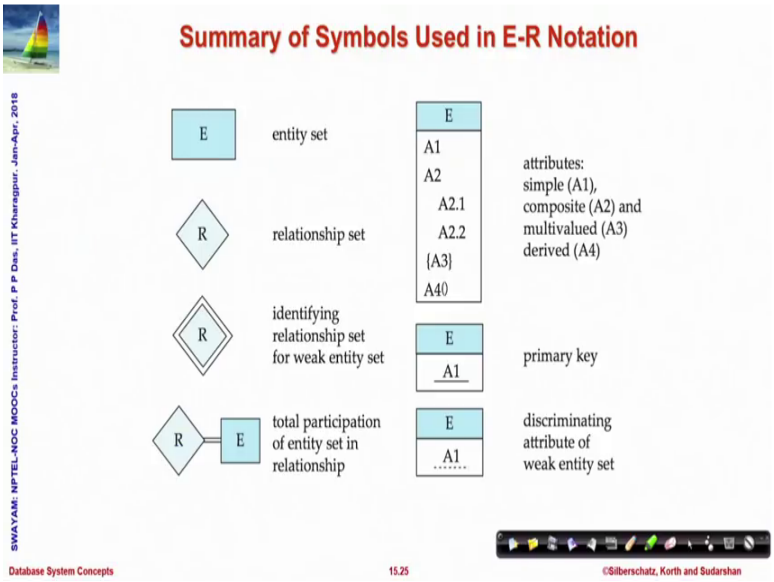
So, if we summarize the discussions on the E-R design decisions; we see that the first decision that we need to take in case of design is the use of an attribute or entity set to represent the object.

So, that is the first modeling that what is the concept and what is what are the attributes or what is the representing entity set for the object that we are trying to deal with instructor, student project and so, on. And we will also have to see whether in the real world this actually is an entity set or it is a relationship set that it is not and concept by itself, but is a concept which relates two or more entity sets and thereby becomes a set of representation.

The use of ternary relationship versus a pair of binary relationship; this trade off will have to be weighed as a design consideration; we have to look into the use of strong or weak entity set. So, we will have to identify the weak entity sets and see if they should be represented through the identifying relation as against a strong entity set. We have to identify the specialization generalization situation where so, that we can get more specific information and create appropriate modularity in the design.

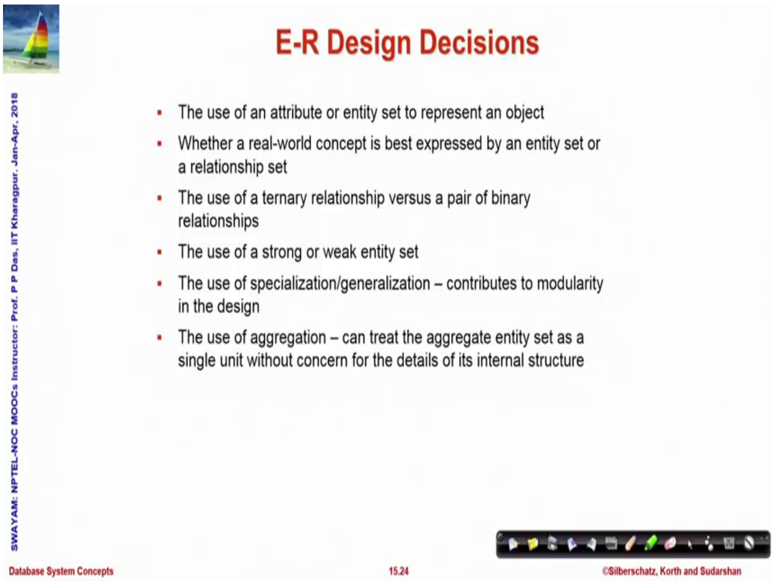
We have to look at aggregation which where we can aggregate entity sets bound by a relationship and create an abstract single unit which can play a role of an independent entity set in the whole design.

(Refer Slide Time: 34:00)



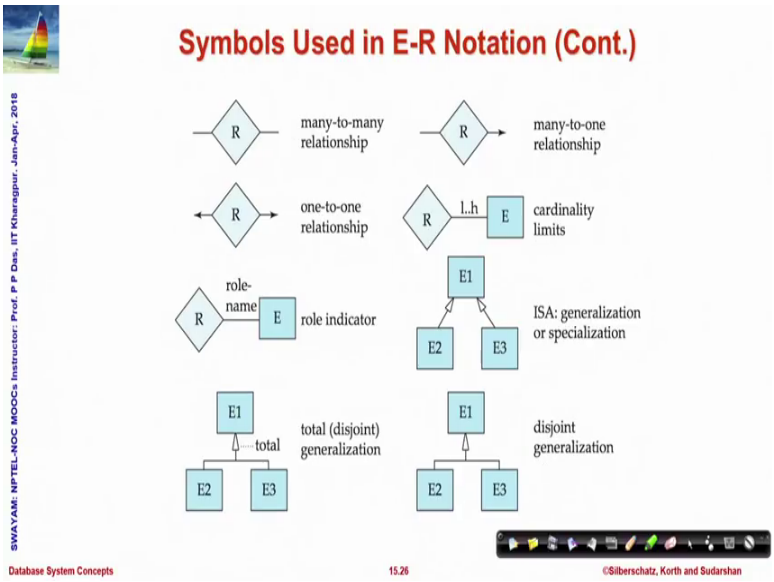
So, this is the basic. So, these are the basic design decisions that you need to make.

(Refer Slide Time: 34:02)



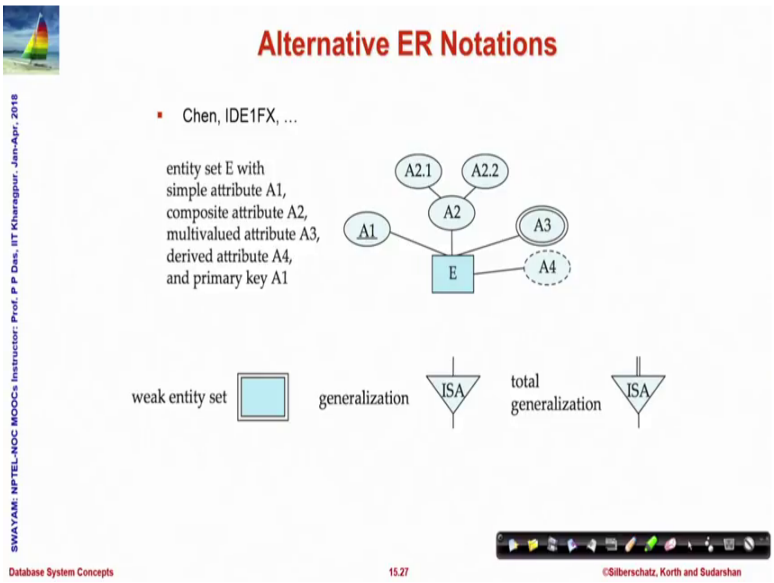
And we will certainly come up with lot more of design decisions as we go along. And before I close in the presentation I have summarized the different symbols that are used in the E-R notation. So, I will not these have already been discussed in depth. So, I will not go through them one by one, but I have put them as a list in the couple of slides.

(Refer Slide Time: 34:30)



This is a next slide in that which will be a quick reference for you while you are initially doing the E-R diagram so, that you know exactly which symbol to pick up for what situation.

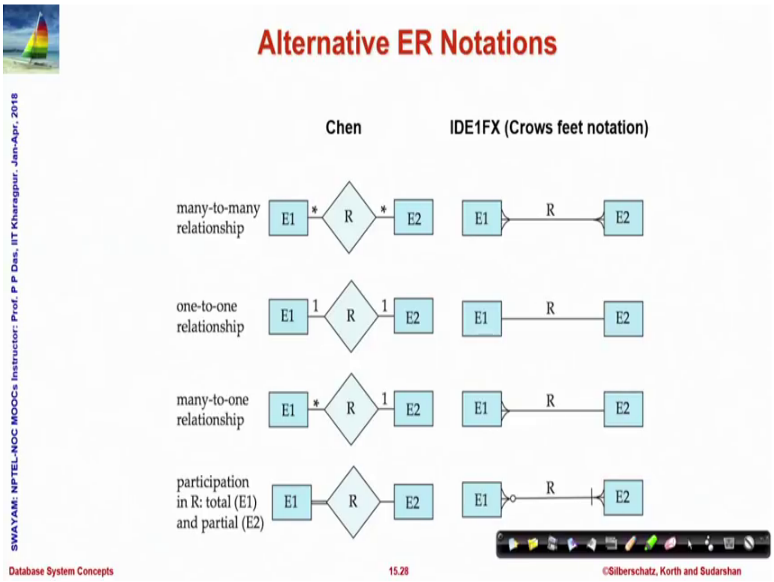
(Refer Slide Time: 34:42)



And at the end also there are few slides which show you that the E-R notation itself is not a unique one.

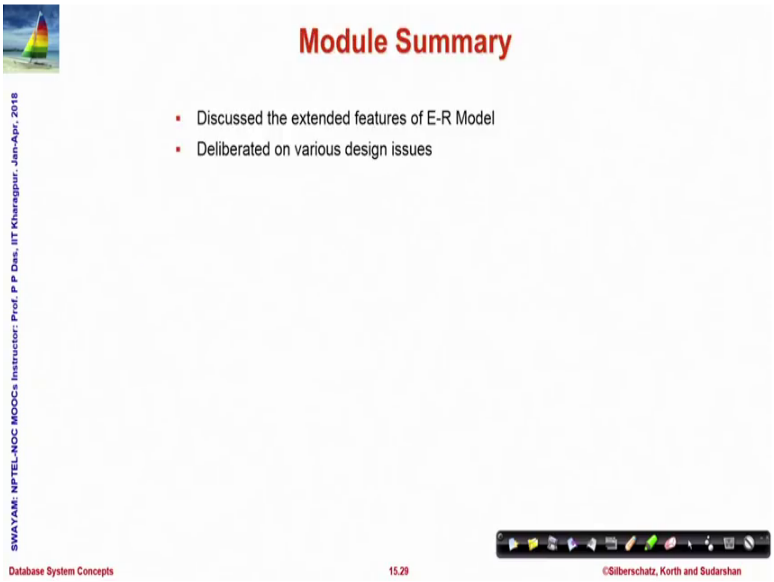
There are multiple ways to represent similar things for example, this is one which is showing you different composite attributes, the generalization, relationship is shown differently.

(Refer Slide Time: 35:04)



So, there are; these are all different styles of showing the constraints that that apply to a particular relationship. And we will I mean we have included this not because we will use these alternate notations, but I have put them because it is possible that you come across some E-R diagram where these notations are used and if you come across and you are not able to identify then please refer to this slides.

(Refer Slide Time: 35:36)



And you will be able to recognize what is what is corresponding symbol that you already know.

So, in this module we have discussed the extended features of E-R model and we have deliberated on certain design issues. And we will close our discussion on the entity relationship model here and move on to discuss the actual relational design.