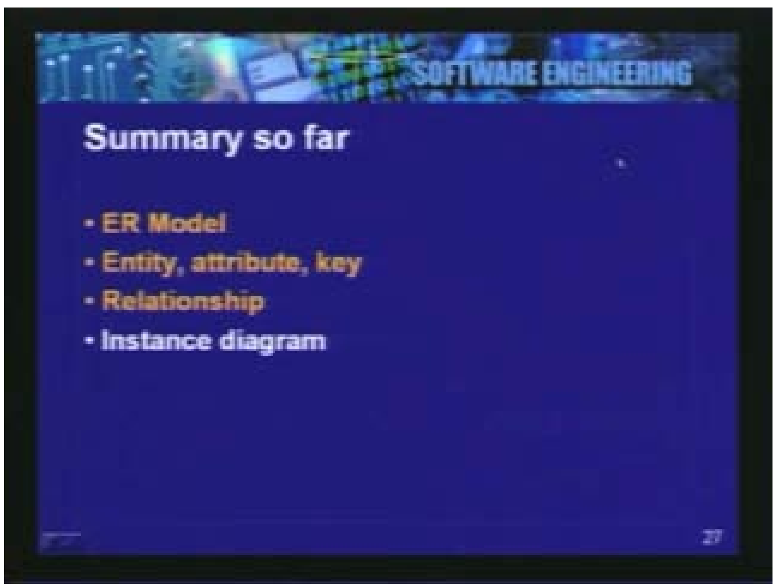
Software Engineering
Prof .N. L. Sarda
Computer Science & Engineering
Indian Institute of Technology, Bombay
Lecture - 10
Process Modelling
DFD, Function Decomp (Part 2)

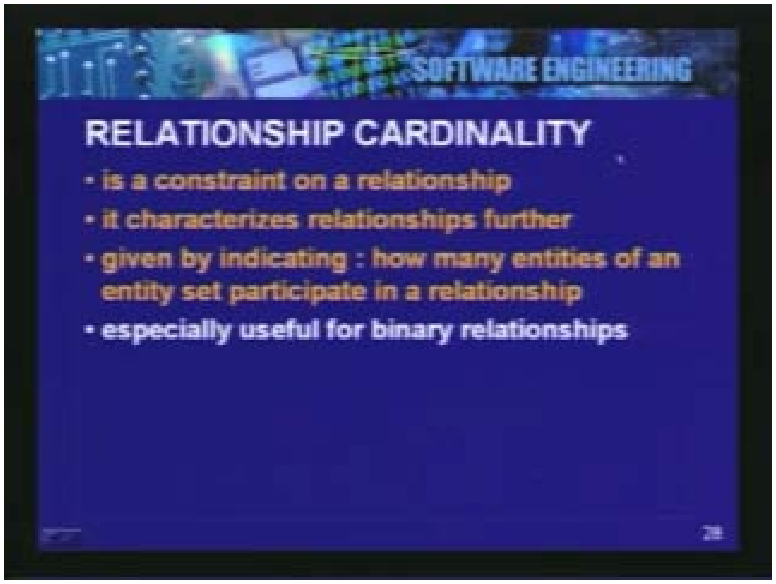
Let us continue with the data modeling topic. So far we have seen the basic features of ER model, where to use it, what advantages it offers. We studied the concept of entity, attribute and the key attributes. We also looked at the relationship concept. And then we looked at a simple diagramming notation through which you can show some entities, some relationships among them. This is what we call an instance diagram and it is always good to draw an instance diagram to illustrate the understanding or to illustrate the concepts we are modeling in our diagram.

(Refer Slide Time: 1:37)



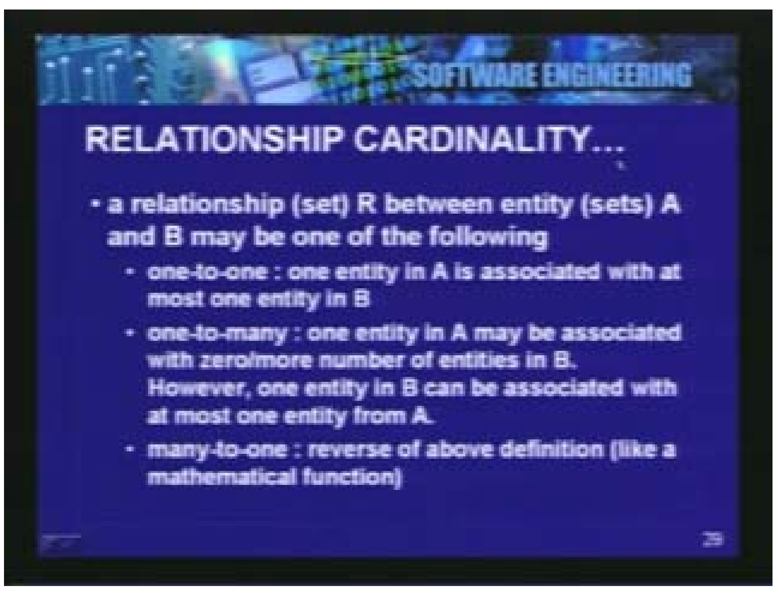
Let us proceed further. We were talking about relationship cardinality as an additional constraint on a relationship. It tries to characterize the relationship more precisely by specifying how many entities of one set may relate with another entity in another set as a part of the relationship. So primarily by cardinality we are indicating how many entities in terms of one or more are related to entities from another set in forming a relationship.
This is a very useful concept for binary relationships where two entity types or two entity sets are related.

(Refer Slide Time: 2:43)



If you consider a relationship named R between two entity sets A and B, then what are the possible cardinalities that we can indicate. The first one is of type one-to-one. It means that one entity in A is associated with at most one entity in B. Here it is a one-to-one relationship.

(Refer Slide Time: 3:13)

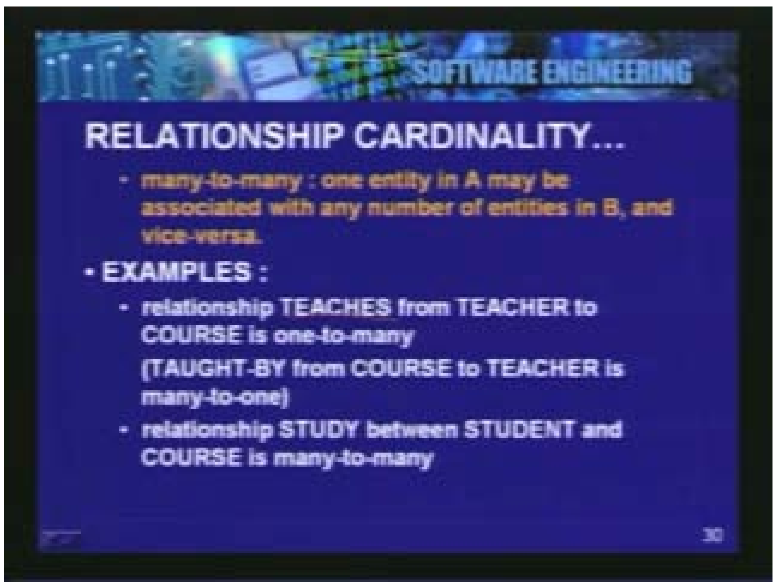


One-to-many relationship is where one entity in A may be associated with zero or more entities in B. When we go from A to B we may meet many entities from B which are associated or connected through the relationship with the given entity in A.

One-to-many is also an indication that if we start traversing or navigating in the database when we start from A we may reach multiple instances of entities in B. This is characterized as one-to-many relationship. Obviously in a given relationship like this, if you now go from B to A, then from one entity in B you would reach at most one entity in A. One-to-many and Many-to-one are really the inverse of each other.

Then finally you have the many-to-many relationship. In the many-to-many relationship one entity in A may be associated with more than one entity in B and also the reverse. Here one entity in A might be reachable from many entities in B and similarly an entity in B may be reachable from many entities in A. This is a many-to-many example of relationship cardinality. Let us look at some of the examples take the example called teaches. The 'teaches' relationship exist from entity teacher and course.

(Refer Slide Time: 5:18)



We have a teacher entity and we have a course entity. Between them we have the TEACHES relationship. What is the cardinality of the TEACHES relationship? Here we are saying one-to-many. That is one teacher may teach many courses. But if you go from course to teacher, then it would be for a given course there is one teacher. One-to-many indicates that a teacher may teach many courses whereas the course is always taught by one teacher. Obviously as you can see here, the relationship is giving you more information. It tells you about how many entities are relating to each other in a given environment. The TEACHES relationship is from teacher to course we said, that is how we understand it from the readability point of view.

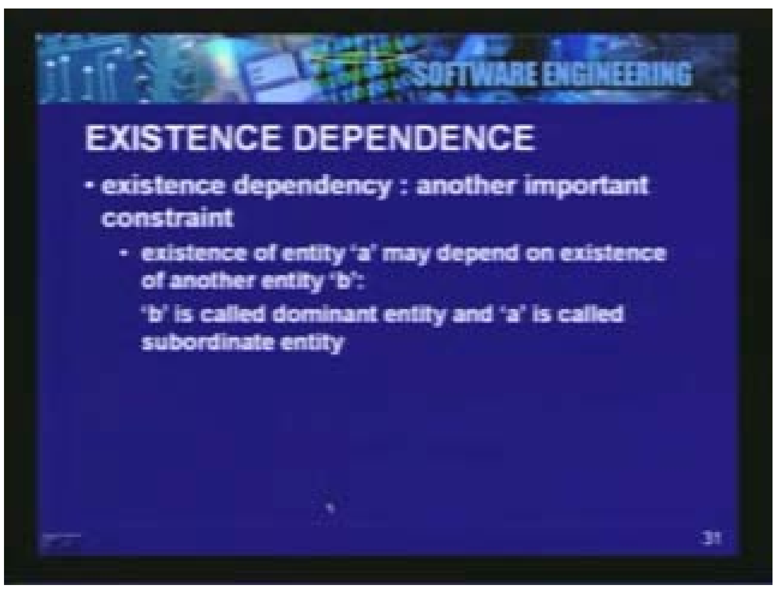
The inverse of TEACHES relationship, that means when you want to go from course to teacher, you may refer to name it as 'taught by'. But in the ER diagrams that we will draw a relationship carries only one name, whether you go from one entity to another or from the other entity to the first one. So the relationships will have the same name irrespective of the direction you are traversing the relationship.

The other example is the STUDY relationship. The STUDY relationship exists between student and course. What is the cardinality of this binary relationship? A little thought will tell us that it is a many-to-many relationship. What it means is that one student may be studying many courses and one course may be studied by many students.

Again if you draw an instance diagram and you show a few courses and a few students then, if you start from one student you would reach on the other end multiple courses. Similarly if you start from one course, you may find many students who are studying that course. So this is a many-to-many relationship.

We can characterize relationships further by indicating the cardinality. It is a very useful concept and you must specify this at least for the binary relationships that you define in your data model. The next concept is that of existence dependence. Again this is a constraint we specify this in order to indicate what is permissible in our data model. It is an important constraint what it specifies is that existence of an entity A may depend existence on another entity B. That means although their entities in their own right, they are distinguishable from each other. So the entity still has the same definition that we gave earlier, that they are unique and they are distinguishable.

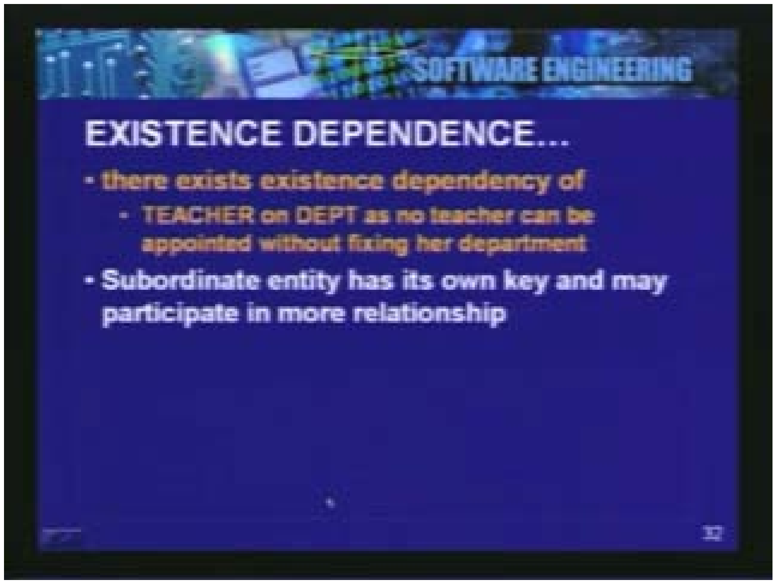
(Refer Slide Time: 8:15)



But in our real world we are saying that entity A must exist in association with entity B. This is called existence dependence. In this case B is called dominant entity and A is called subordinate entity. An entity which depends on other entity for its existence is naturally called a subordinate entity. In this case both the entities are proper entities, they will have their own attributes, and they will have their own keys. But still when they are created, when they exist in the real world, they exist in association with another entity. A simple example would be the dependency between Teacher and Department. Teacher and Department are two entities and they have a relationship between them which we may call as employee.

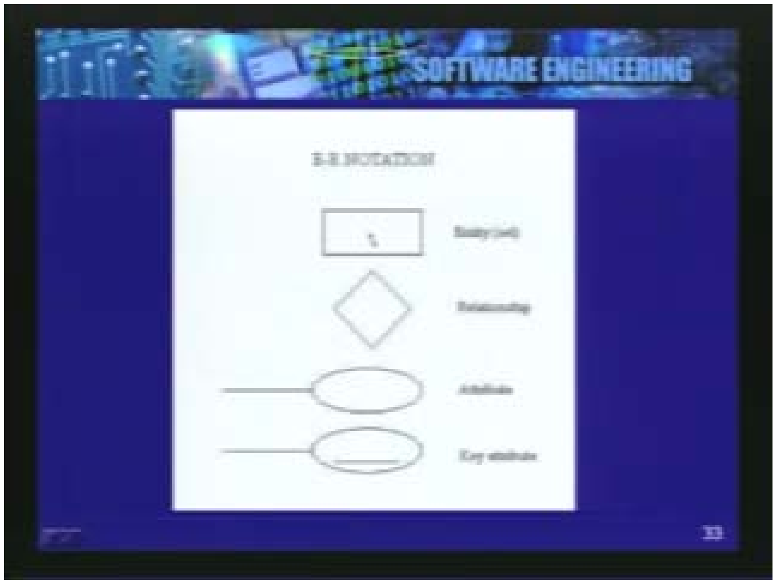
So we say Teacher is an employee in some Department or Teacher is employed in some department. Whenever a new teacher is appointed, the teacher has to be appointed in a particular department. What it means is that, teacher cannot exist without being associated with some Department. So Teacher depends on Department and hence there is existence dependence between these two entities, where the Teacher is the subordinate entity and the Department is the main entity.

(Refer Slide Time: 10:23)



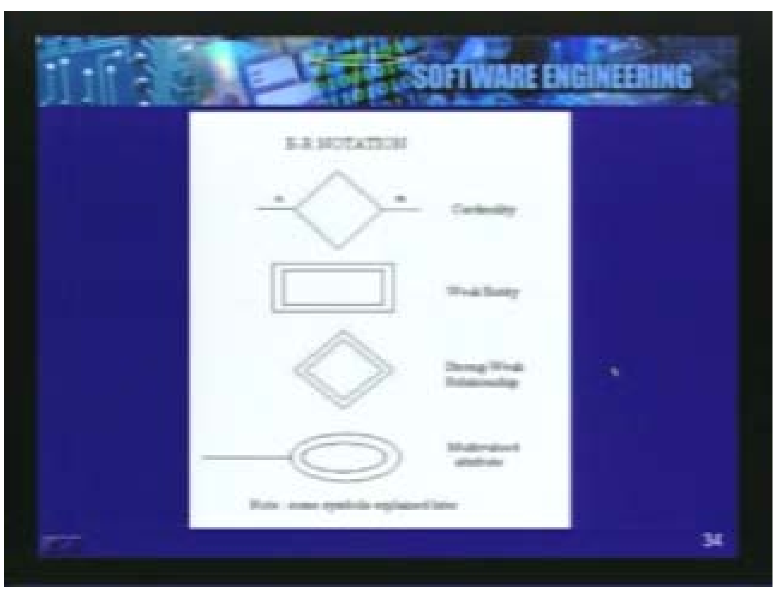
Again I am emphasizing this that the subordinate entities do have their own key and they can participate in additional relationships. They are entities in their own right, we are only expressing the real world requirement that an entity like this must exist in association with another entity. Let us now look at the diagramming notation for the ER model and look at some more examples. Here are the diagramming notation symbols we will be using. An entity is represented by a rectangle and we will write the name of the entity within this rectangle.

(Refer Slide Time: 11:10)



A relationship is represented by a diamond. These are basically notations for the entity set and the relationship set. An attribute is written down in a small circle which is connected either to an entity or to a relationship, because attributes do not exist independently. They are associated with entity or relationship and the key attribute may be underlined. Here are some more notations. The cardinality of the relationship is indicated by writing a number. Usually we write ‘one’ or ‘many’ or ‘n’ to indicate many.

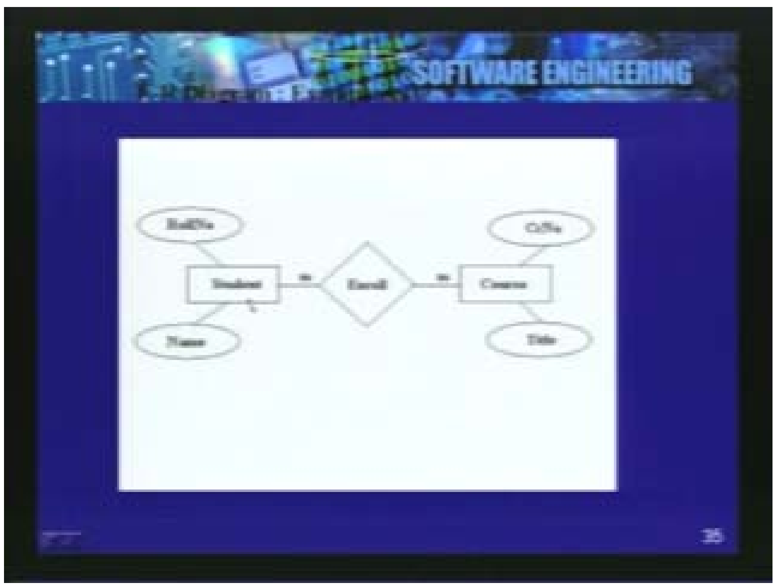
(Refer Slide Time: 11:49)



We may say the cardinality is many-to-many. In that case we may write a letter ‘n’ in one side and letter ‘m’ on the other side. Or we may write the letter m at both the places. This is how the cardinality for binary relationships is shown. These two symbols are used when we encounter weak entities. We have not yet seen those concepts, so we will see their usage in more details subsequently. A multi valued attribute is shown by a double circle. Recall that we had taken example of such an example such as the author of a book. Author will be written in a double circle and would be connected to the book entity to show that this is a multi valued attribute of the book entity.

These are the diagramming notations so there is no universal standard for these notation but these are the symbols which are very commonly used. Let us look at this simple example. We have shown here an entity called student. This entity has two attributes roll number and name. We indicate entities and we associate attributes in this fashion. We may also say that roll number is the key for the student entity.

(Refer Slide Time: 13:21)

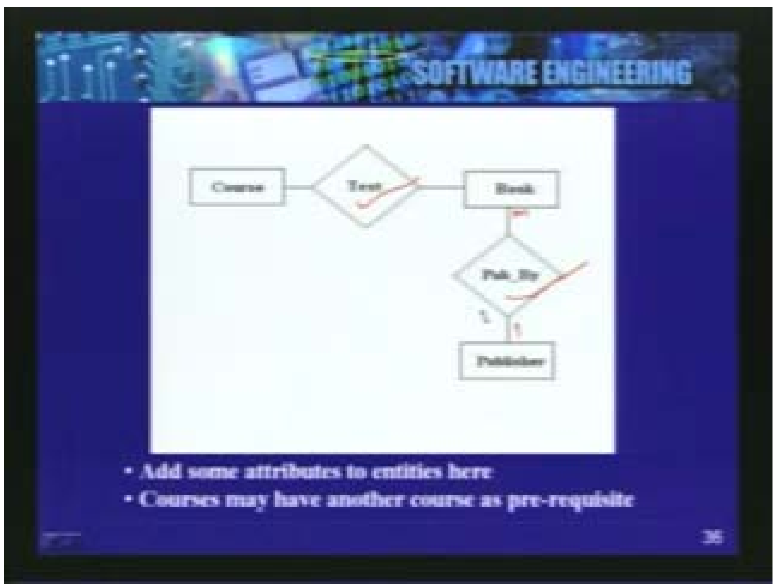


Another example in this is the course entity which has again two attributes, the course number and the title. The course number may be the key for this entity. These two entities are related through the enroll relationship; student enrolls in a course. This enroll relationship has the cardinality many-to-many. A diagram like this you can see is simple to read and it is also precise and it represents the real world by defining appropriate constraints. For example we have identified a binary relationship called enroll, we have also defined the cardinality, we have identified the keys, and we have defined the entities. This diagram is very precise, it is unambiguous, and it can be read by many people with the same understanding. There is no different interpretation of diagram like this. This is one of the advantages of ER diagram.

And the other advantage is that once you have prepared an ER diagram and if you have named the entities and attributes well, anybody should be able to read and understand the diagram. Let us look at some more examples. In this example we have a few entities and relationships. It is best to read the diagram and first understand the entities.

We have a course entity; we have a book entity and a publisher entity. From the names of these entities we have a clear understanding of what they could be implying and what this model could be for. Obviously we are talking here about books published by different publishers and which are described as text for various courses. The connection between the entities that we have here is through two relationships; the 'Text' relationship and the 'Published by' relationship.

(Refer Slide Time: 16:10)

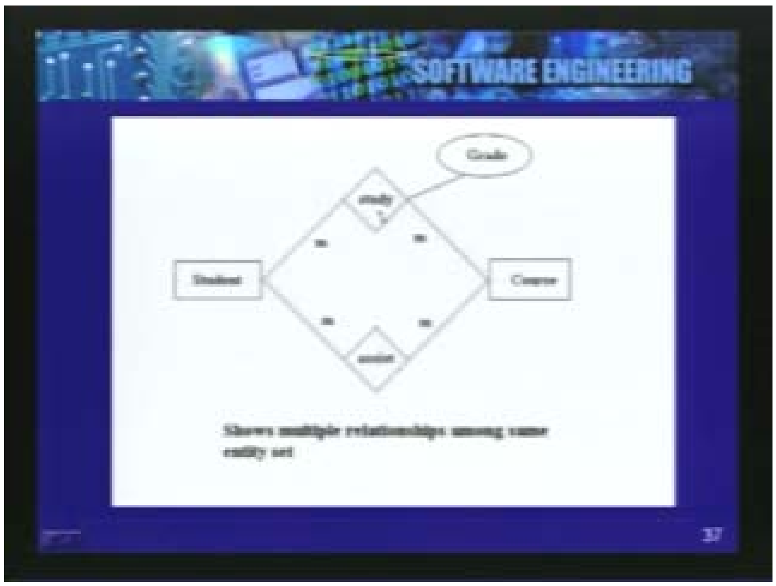


We have three entities and two relationships among them. Obviously this diagram is not complete. You need to add a few attributes here which are relevant to your application. Then you need to characterize the relationships further and the cardinalities. Consider the relationship 'Published by' between book and publisher. What should be the cardinality? What should we indicate on the 'Book' side and what should we indicate on the 'Publisher' side? If I go from the 'Book' towards 'Publisher', how many publishers am I likely to encounter? Usually all books have one publisher. I would generally put '1' as cardinality on the 'Publisher' side and I will say if I go from 'Book', I am likely to meet only 1 'Publisher'.

But if I go from 'Publisher' towards 'Book', naturally a publisher publishes many books. So I may put 'm' (many) here to indicate that there are more books published by a single publisher. This is our understanding of the real world. Of course in the real world it is possible that a book may be published by more than one publisher and they may sell it in different market segments. In that case this characterization of 'Published by' relationship may not be appropriate and we may have to make it many-to-many.

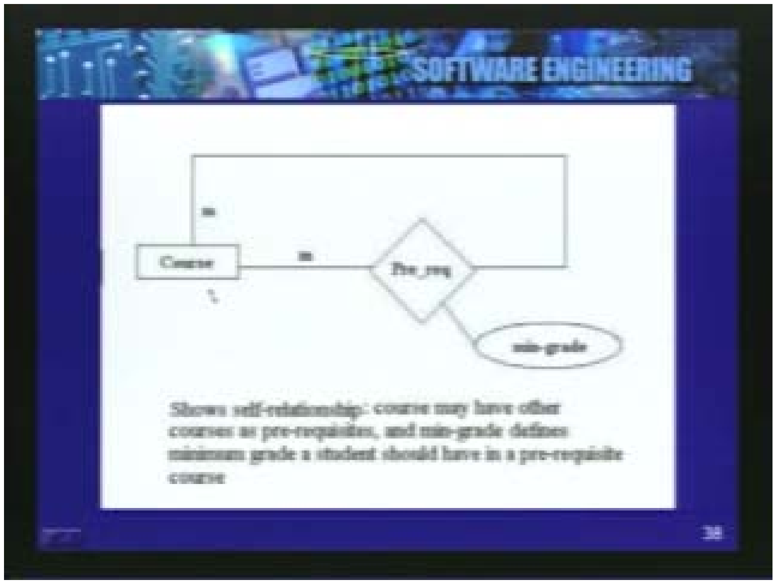
What it really means is that the model has to clearly reflect the real world and ER model provides you enough facilities by which the real world can be correctly captured. We will leave this as an exercise for you to complete this diagram. Here is another diagram from our familiar university environment. It shows two entities students and course and it shows two different relationships between the same two entities. We have students who study different courses and we have students who assist in different courses. This study and assist are entirely independent relationships. Such multiple relationships may exist among the same set of entities and the attribute ‘Grade’ is an attribute of the study relationship.

(Refer Slide Time: 18:53)



Here is an interesting aspect of the modeling where we are showing self relationships.
We have a course. A course may have a prerequisite course which means that the course is actually related to other courses which may act as prerequisites for a given course. This is a self relationship what we are saying is that a course is related with itself through a relationship called prerequisite.

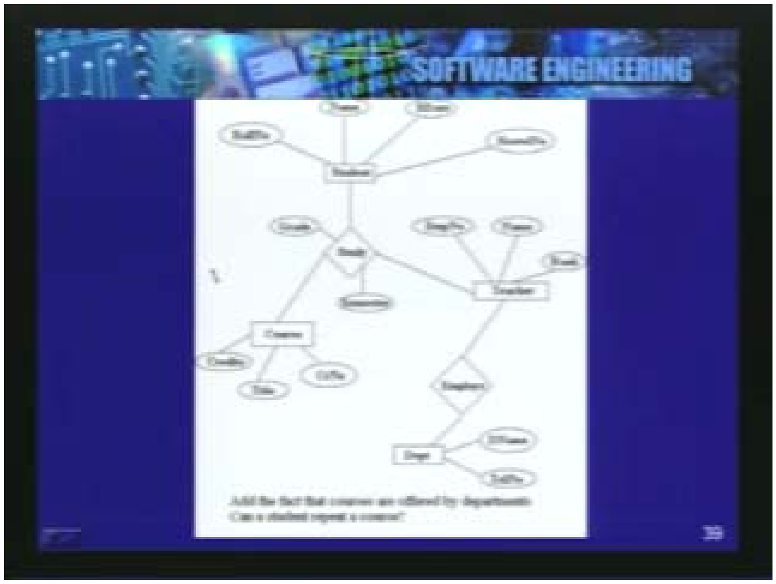
(Refer Slide Time: 19:21)



This relationship is also binary relationship. Let us find out the cardinalities here. A course may have more than one prerequisite. For example a database course may have data structures as a prerequisite. It may also have one more prerequisite say programming languages. So a course may have more than one prerequisite. Similarly a course may be prerequisite for more than one course. Hence in both the direction it is many-to-many. Such self relationships can be shown in the ER diagram and we can even have some attributes associated with such relationships. For example: what should be the minimum grade a student should get in a prerequisite course, so that he can study a follow up course?

This can be captured by the attribute minimum grade associated with the prerequisite relationship. Let us look at little more complex diagrams which are giving a more larger scenario and which are modeling entities and relationships identified in that particular scenario. The point that we made earlier, given a diagram like this you should all be able to read and understand in exactly the same way as I would read and understand them.

(Refer Slide Time: 21:06)



Let us try to understand these. We can first look at entities shown here and try to get a good idea of what they may represent in the real world.

- There is a student entity which has some well known attributes: roll number, name, date of birth, and the hostel number.
- We have the course relationship, which is identified by course number, title and has credits associated.
- We have teacher entity.
- We have department entity.

We have four entities here. Having understood the entities and also the attributes that they have, let us look at one relationship at a time. We see here the relationship which says department, employees and teachers. Again it is obvious to understand because the name ‘employees’ conveys the connection between the two entities. Here we have a ternary relationship. This ternary relationship says that student study courses under teachers. A STUDY relationship associates a teacher, a course and a student and it has attributes ‘grade’ and the ‘semester’.

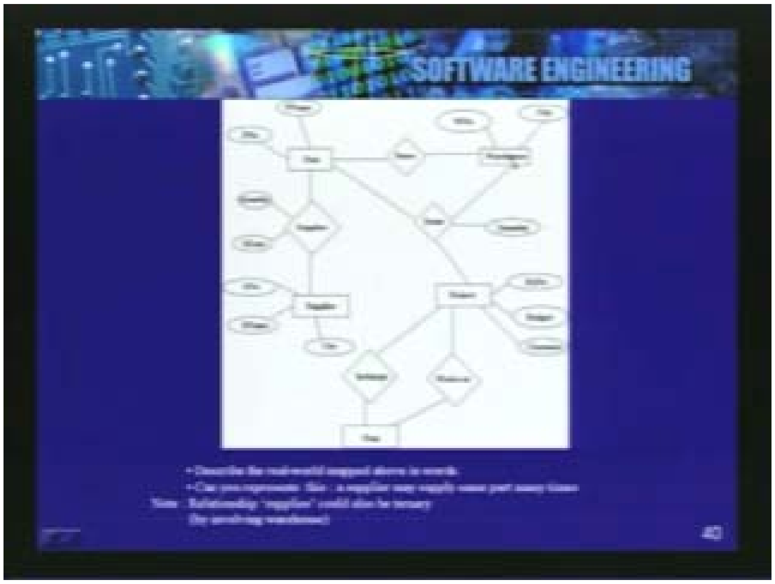
This means, a particular student may be studying a particular course under a particular teacher in a given semester and he would get a particular grade. We are able to capture the study aspect of the real world by a ternary relationship between student, teacher and course. What would be the key of this relationship? We had said earlier that the key of the relationship is the combination or concatenation of the keys of the participating entities. The key of study would be ‘roll number’ from student, ‘course number’ from course and ‘employee number’ from teacher. The three of these attributes together would form the STUDY relationship’s key.

This allows us to model a real world where same course may be run in the same semester by different teachers. When we model this we have to be very careful whether this model takes care of the real world requirements. Let us take another example. Does it allow you to represent a case when a student fails in a course and he studies the same course again may be with the same teacher but with different semester? Yes. It happens a student might fail and get a grade let us say an ‘f’ grade by a fail grade and then he has to register for the same course again. The next time he may have the same teacher. Does this model allow you to capture this kind of situation? I will leave this as the exercise to be taught over. We will come back to this version later in the context of another example that I will give you.

This is another useful example and again you should be able to read and understand this diagram. I will quickly tell you the entities which are shown here. I have the ‘supplier’ entity and I have the ‘part’ entity and naturally the supplier supply these parts. This is a standard example where we need some parts and we identify some suppliers and we know who supplies what parts. We may receive some parts from a given supplier in a given quantity and on a particular date. Let us say supplier S1 may supply to us a part called P1 in quantity hundred. This is an example where we buy or we receive parts from different suppliers.

Then we have different projects and we have warehouses. The parts are stored in different warehouses and projects need these parts and these parts may be supplied to projects from different warehouses. It is possible that a part P1 is given to project from a warehouse W1.

(Refer Slide Time: 29:28)



The same project may get the same part from another warehouse also. This is shown as a ternary relationship between Part, Warehouse and Project. It has a specific implication and it is trying to model the real world. Our idea in modeling should capture the real world as faithfully as possible. The best way to verify whether the model is correct or not is to take some examples and see whether those examples are clearly possible with respect to the model that we have put here. Then we have employees who are working in the projects and one of the employees may act as an in-charge of the project. This is the ER model for a company which might be managing project and which needs certain parts which are supplied by different suppliers.

We will again look at some specific issues in this model and check whether it does what I need in my real world, whether we have done it correctly or not. Let us concentrate on the part on the left hand side of this diagram and we will discuss some issues and see whether those are handled properly. We say that the supplier may supply part: What is the cardinality of this relationship? It is obviously many-to-many. A supplier can supply many parts and a part may be supplied by many suppliers. So we can call it as a many-to-many relationship.

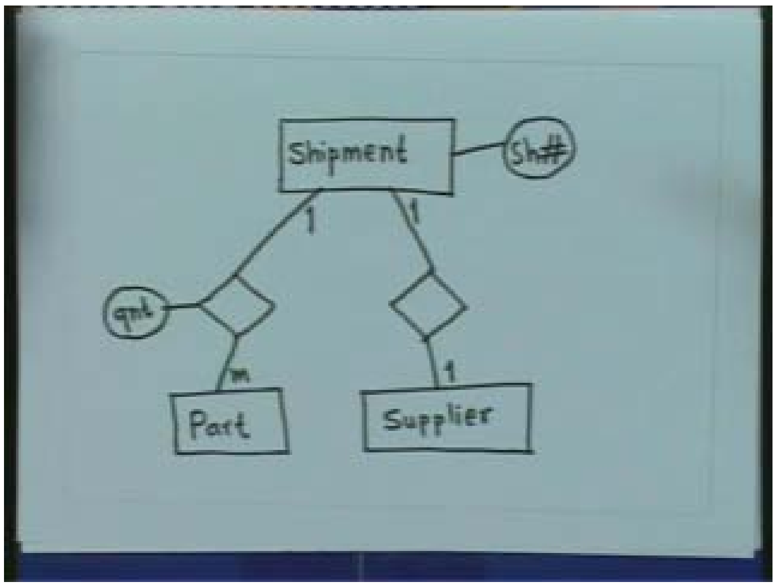
In this case, let us ask the question: Can the same supplier supply the same part twice or three times may be on different dates and in different quantities? This is something which is quite possible in the real world. We may find that a particular supplier is best suited for a particular part and we may repeat orders with him. Can the same supplier supply the same part again to us? Is this covered by the model here? Again to answer to this question, you must clearly understand what this relationship means and what the keys of that relationship are. When we say that the relationship ‘supplies’ is identified by the key of the supplier and by the key of the part, it means that I cannot have a situation or I cannot have another instance of the relationship supplies with the same key that means the same supplier and the same part.

What it means is that, this model therefore cannot represent a multiple supply from the same supplier for the same part. This may not be appropriate and this may not be what ever real world needs. We need to modify this. When we discuss this with the user, user would indicate that this part of his requirement or this part of the real world is not captured by the model correctly. We need to think about how to modify this model. In fact, in a way this problem is related with the problem that we discussed in the previous diagram where we talked about the student repeating the same course with the same teacher possibly in another semester.

Once we try to understand the problem, we will be able to come up with the solution. In both the cases you see here that the two models that we have are not handling the situation that we want. Basically we want to distinguish between the different supplies made by the supplier. Supplier may supply parts, each one of them is a shipment by itself. We need to distinguish the shipment on its own rather than distinguish it through primary keys of the entities which are participating. What it means is that, we need to distinguish the instances of the relationships on their own and therefore it cannot be modeled as a relationship.

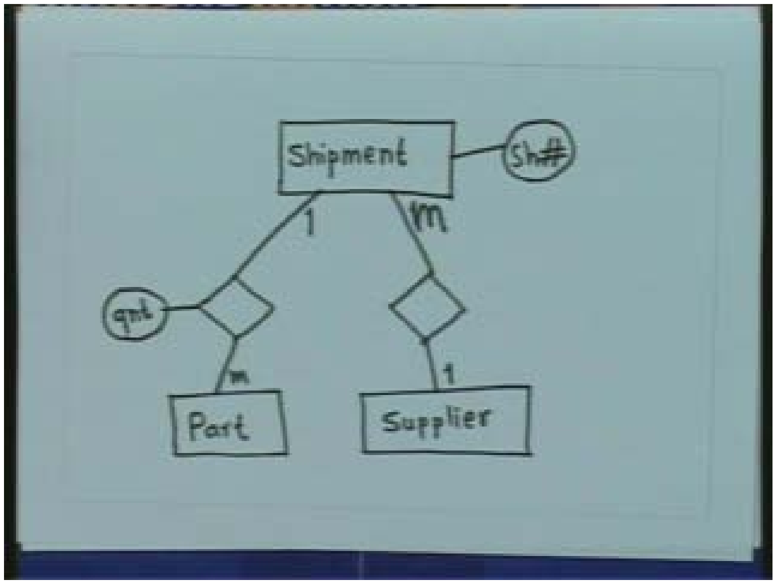
We need to think of this in a different way which will allow us to distinguish one shipment from another shipment irrespective of what it contains and where it has come from. We need to change this part of the diagram so that it captures this requirement. I have already given a hint that we need to create an additional entity because entities are distinguishable in their own right and by creating a separate entity we should be able to address this. Here is a solution we can propose for the part which we did not find satisfactory. As you see here, we have created an entity called 'shipment' and this entity has its own key called shipment number. With this key we can distinguish instances of shipment from each other.

(Refer Slide Time: 32:43)



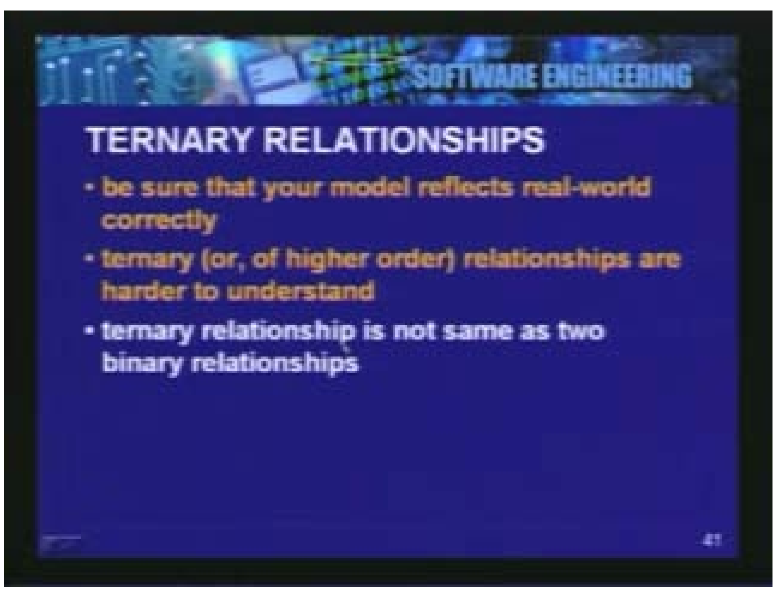
I have now two relationships. I say the shipment consists of parts and shipments are obtained from different suppliers. Shipment is now having relationships connecting to the part and connecting to the supplier. What should be the cardinalities here? Obviously a shipment can consists of multiple parts and each part can have a quantity associated. Similarly a shipment comes from a supplier but we are assuming that it always comes from a single supplier. If I go from shipment to supplier the cardinality should tell me 1. But if I go from supplier to shipment is mentioned as 1, which in fact is not correct. Supplier can have multiple shipments. So ideally I should make it as a relationship of type many on the side of shipment. Supplier to shipment is of one-to-many type, but from shipment to supplier it is many-to-one. So we have corrected this situation and now our model is quite precise and it allows us to handle multiple shipments irrespective of what they contain and who they come from. Let us now look at ternary relationships in some more details, although in previous examples we have already encountered them.

(Refer Slide Time: 34:24)



We need to ensure that we model the real world correctly and when trying to do that some relationships may appear to be binary, a few may appear to be ternary or even more complex. They may be four way or five way and so on. Ternary or higher order relationships are harder to understand. You must verify that the given situation demands a ternary relationship.

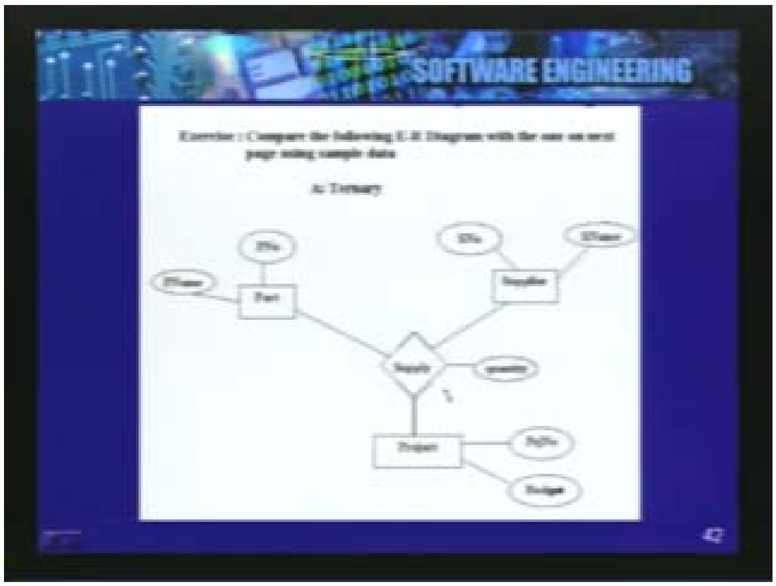
(Refer Slide Time: 35:50)



We had taken the example of STUDY earlier and we had justified the STUDY relationship should be a ternary relationship involving teacher, student and course because courses are taught in divisions concurrently.

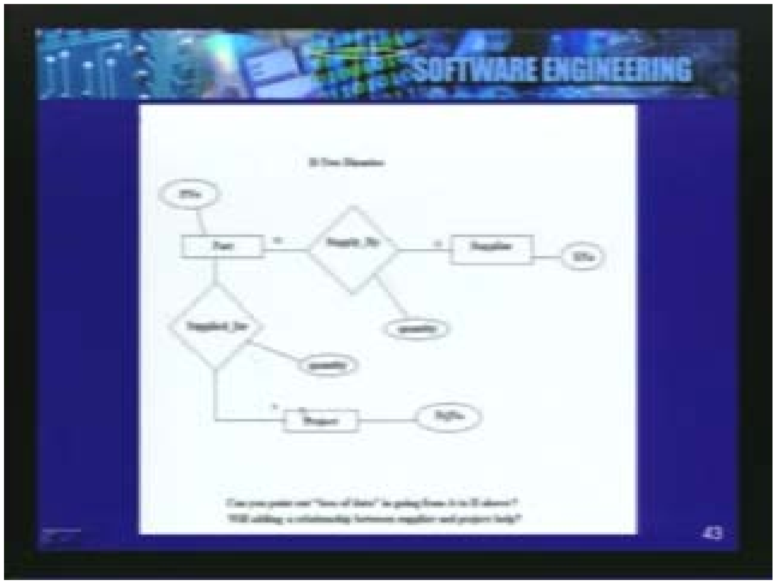
Here the ternary relationship is justified. Ternary relationship is not same as two binary relationships. In fact this is an important point and therefore it has an implication on modeling. Here is an example which shows the ternary relationship between part, supplier and project. Let us assume that this reflects my real world correctly. That means parts are supplied to projects from different suppliers. So we have shown it as a ternary relationship.

(Refer Slide Time: 36:38)



Can I show the same thing using two binary relationships; one between part and project and the other between part and supplier as has been done in the diagram B? In fact, to verify this, you again should try to walk through the diagram. Let me take one project. From this project, I can go by this relationship and I will find out all the parts which have been supplied or which have been received for this project. So I can go to part say P1 by P2 by P3. Once I have reached P1 by P2 by P3, I naturally forget from where I have come, because I am only at the part instance. If I take up P1 and follow through, I can find out who are the suppliers, all the suppliers who have supplied this part P1. But because I have broken it into two parts, I cannot say this is the supplier who had supplied for the given project. If I start from that supplier I may go to many project parts. I take one part and I come here and I reach many projects. In general I will not be able to now tie down which part was supplied in what quantity for which project.

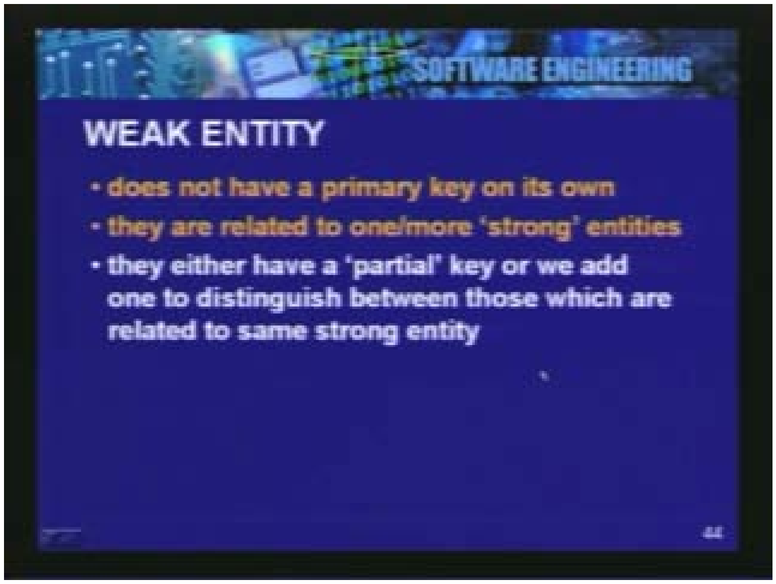
(Refer Slide Time: 37:12)



If we are talking about the same real world which was shown earlier as a ternary and now it has been shown as two binary, then it is not the same. One of these has to be incorrect. If the ternary relationship is the correct one, then replacing it by two binary is a modeling error and it is not the same. This is the case even when you add additional binary relationship between project and supplier, it will not capture the main point that we are making. The question, what was supplied, in what quantity, in which project and by which supplier is properly answered by the model that we used a ternary relationship. But it is not properly handled by the two binaries.

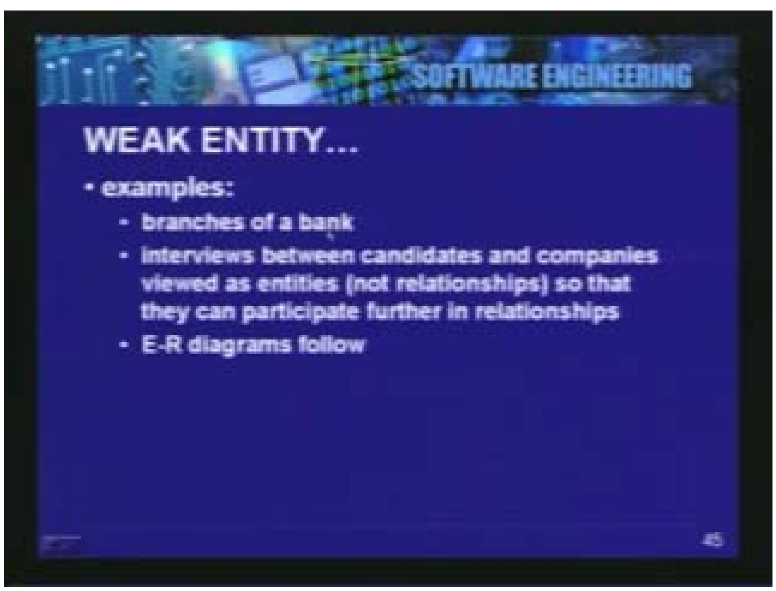
Let us now go to the next concept, the 'weak entities'. There are times when you encounter entities which do not appear to have a primary key on their own. Such entities are called weak entities and they are always related to some other entities which are normal entities, which we may call it as strong entities which have their own primary key. These entities which do not seem to have primary key attribute on their own seem to exist in the context of some other strong entity. We will see some examples. Such weak entities may be distinguished in the context of a strong entity but not on their own. This is a situation where we have to use the concept of weak entity. Here are a few examples. The first one is about the branches of a Bank. Let us consider a Bank like State Bank. State Bank is a strong entity because it has a name 'State Bank' and no other Bank can have same name. So you may have State Bank, you may have Canara Bank.

(Refer Slide Time: 40:34)



These are the names of the Bank which distinguish one from the other. So Bank is a strong entity. But Banks have branches and how are branches distinguished? Most Banks give some serial number or name to the branches. State Bank may have a branch whose name is IIT Powai branch; similarly Canara Bank may also have a branch whose name is IIT Powai branch. Branches are on their own, they do not have unique names.

(Refer Slide Time: 41:01)



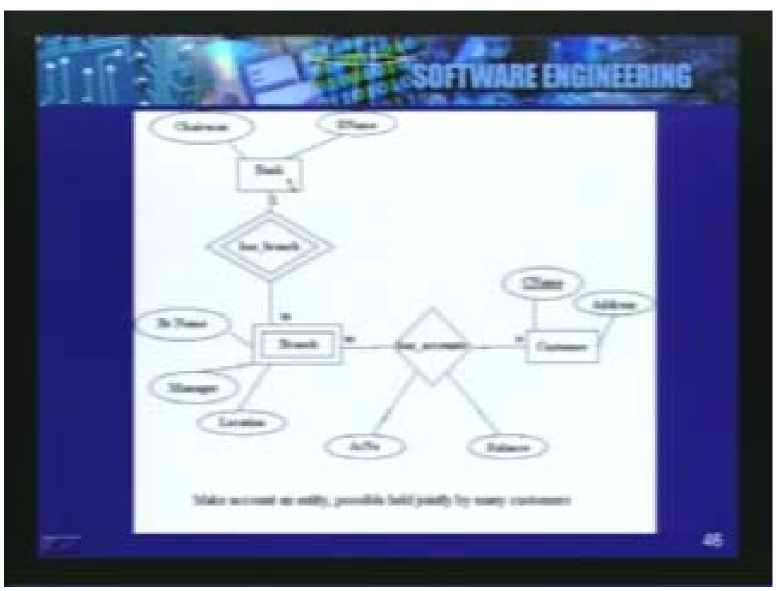
We have both branches called IIT Powai branches, but they belong to different Banks. This branch name does not appear to be the primary key for branches.

Banks may otherwise use some serial numbers and these serial numbers may distinguish branches from each other, in the context of a Bank. We can talk of branch one of State Bank, branch one of Canara Bank. These serial numbers would also be in the context of the Bank. Therefore branch appears to be a weak entity. A Bank may have many branches for example State Bank may have more than ten thousand branches. I need to treat branch as an entity on its own, but it appears to be a weak entity.

Branch should be treated as an entity because, people open accounts in branches not in Bank. You may have an account at IIT Powai branch or you may also have account at another branch of the same Bank and usually customers deal with braches. I would like to represent branch as an entity, but it appears to be weak entity and it appears to exist in the context of a Bank. Another interesting example is the example where candidates appear for interviews with different companies and interviews are important in their own right.

How do I distinguish one interview from other interview? It appears to be weak entity. It depends on the candidate and the company. Why do I want to distinguish an interview from another interview? All interviews may not be successful and only some interviews might be related to job offers. Other interviews may not be related to job offers. If I want to capture this ‘offer’ entity and relate it to ‘interviews’, I must represent interview as an entity. Ultimately relationship exists only between entities. Here you would see that I need to capture interview, show it as entity but it seems to be a weak entity. We will try to now show you how weak entity can be represented in the diagram and we will take both of these examples.

(Refer Slide Time: 44:25)



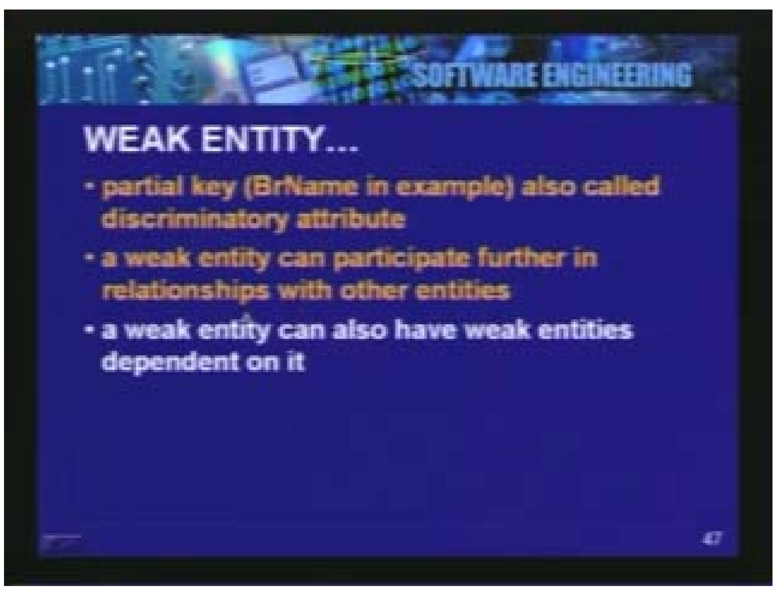
Here is the ‘Bank’ entity it has Bank name and this could be the key and it may have some other attributes. Similarly we have another entity called ‘customer’. These are our normal entities and they are strong entities. They have their own key. Let us for the time being assume that the customer name is unique. We need to introduce branch as an entity.

Branch, as we said earlier is a weak entity. So we show it by a double rectangle. And the fact that branch is a weak entity and depends on Bank, each shown by a double diamond connecting the strong entity and the weak entity. This defines the context, it says branch exist in the context of Bank although branches can be distinguished among themselves. But they can be distinguished only within the context of a Bank and you may have branch name as an attribute which can be used for making the distinction.

This branch can have a relationship called 'has accounts'. A customer has accounts with the branch. A branch may have many customers. A customer may have account with many branches. The customer may have account in say five branches. In order to find out which Banks these five branches belongs to, we have to go actually by this relationship shown in the diagram. Out of this five, two may be with State Bank and another three can be with Canara Bank and so on. That is captured from going from a customer to branch and from branch to the Bank. Here branch is a weak entity.

This is the example where we had 'branch name' as a partial key. It is called the partial key or also called the discriminatory attribute. A weak entity such a branch that we saw can participate in further relationships and we had that example of having accounts with branches. A weak entity can also have weak entities depending on that. There can be a chain of such weak entities.

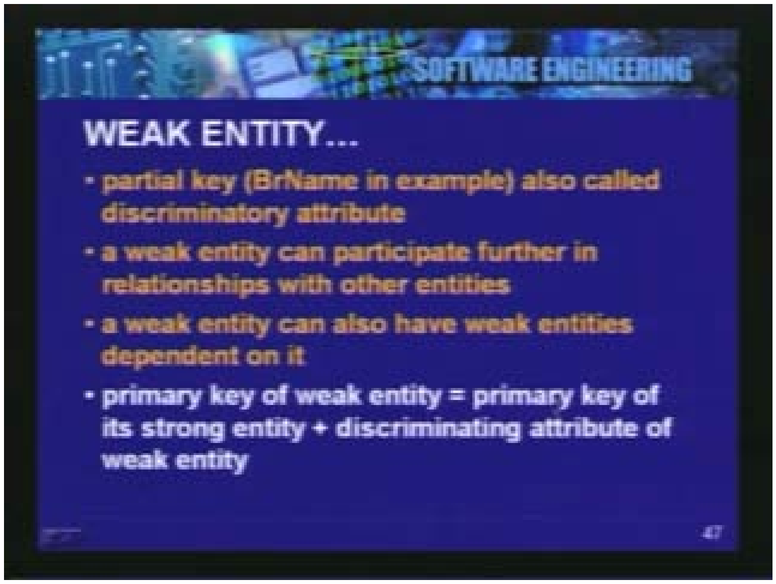
(Refer Slide Time: 47:09)



For example a large organization may have departments. Department is a weak entity. Departments may further contain sections. Section is a weak entity. For example, let us consider, 'Section 1' of 'Department 1' in Organization A. Similarly Organization B may have Department 1. Within Department 1, I may have 'Section 1' and another department may also have 'Section 1'. Hence 'Section' also is a weak entity and you can have a chain of such weak entities and they would exist in the context of some strong entity somewhere.

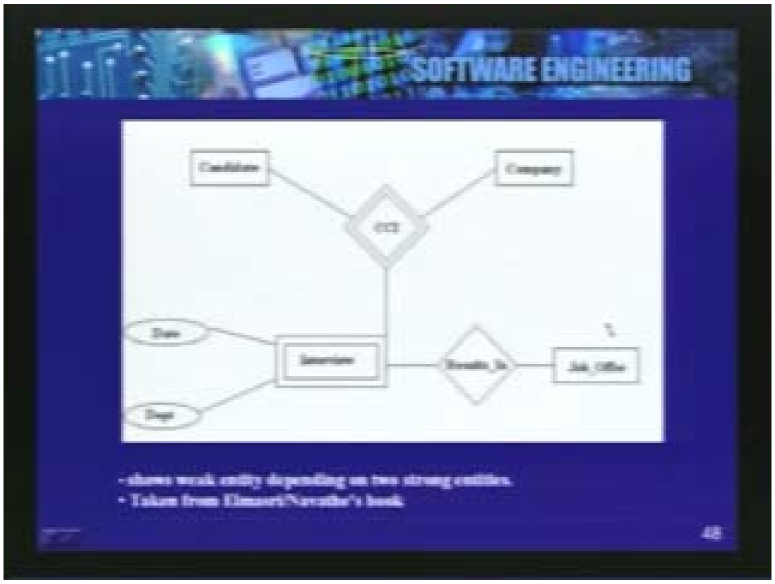
What is the primary key of weak entity? Primary key of weak entity is the combination of the primary key of the strong entity on which it depends plus the discriminating attribute. A primary key for branch would therefore be consisting of 'branch name' and the 'name of the Bank'. These together are unique and this is how we define the primary key for the weak entity.

(Refer Slide Time: 48:00)



Here is the other example which we have taken from this well known book on data modeling and the databases. Here you have the weak entity called 'interview' which depends on the candidate and the company. This weak entity depends on two primary entities and some of these interviews may result in an entity 'job offer'. 'Job offer' may be an entity in its own right. You may have some attributes with it and there may be a key attribute. Assume that we have defined all of them because anytime we draw a strong entity you must clearly conceptualize it. You must always define attributes and you should always answer this question that it must have its own primary key because one job offer must be distinguishable from another offer. For example, the position may be the attribute which distinguishes one job offer from another job offer.

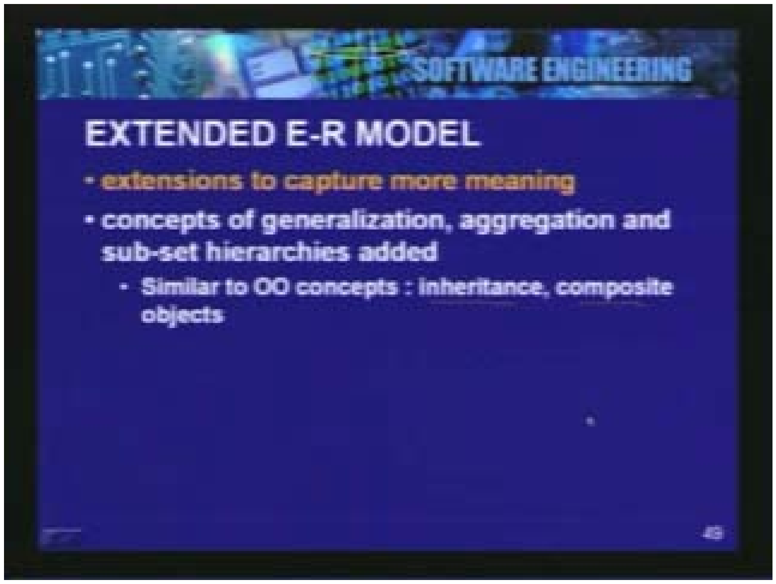
(Refer Slide Time: 49:02)



In this particular diagram can we represent the same in different way and will it convey the same meaning? For example can I have a ternary relationship between candidate, company and job offer? Replacing all of this by a single ternary relationship obviously seems to be wrong because every interview does not result in a job offer. In that case we will not be able to capture those interviews which did not result in a job offer. Therefore we cannot replace this by a ternary relationship. You cannot have a binary relationship between ‘candidate’ and ‘company’ because in that case, we cannot relate that binary relationship with the ‘job offer’. Considering all this you will find that this is the correct model for the situation we were describing.

We have seen the basic concepts so far. We have talked about the most important concepts of entity and relationship. These are the core concepts and then we had other related concepts like attributes, cardinalities, the key attributes and so on. This ER model has been further extended to capture more meaning. Primarily the new concepts which have been added are the concepts which are some what related to concept of the object oriented models. These concepts are the concepts of generalization and aggregation and they also introduce the concept of subset hierarchies. We will see that these concepts give you more flexibility in creating the data model for a given situation and they capture the semantics much more comprehensively.

(Refer Slide Time: 51:43)



The kind of new concept that we will see here are: The concepts of inheritance, which are the concept in object oriented data model and the notion of a composite object or a complex object where an object may contain other objects. So we will study the extended ER model subsequently and see how these concepts facilitate the data modeling exercise.