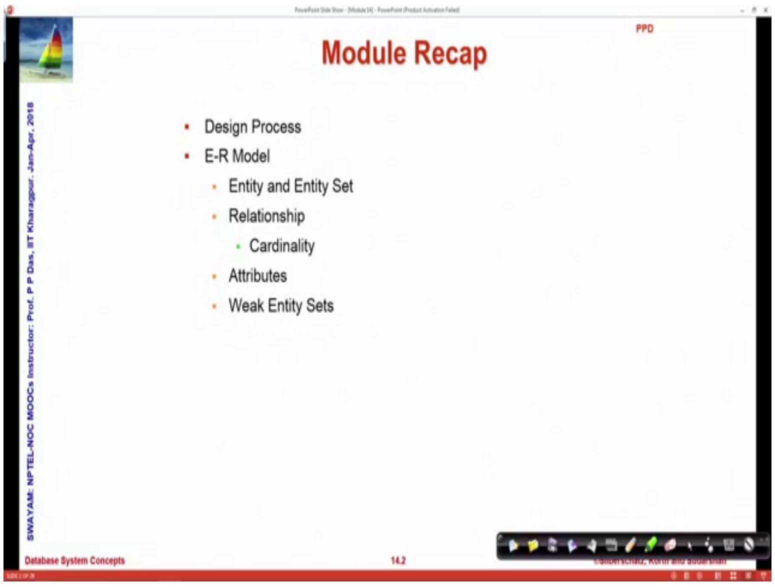
Database Management System
Prof. Partha Pratim Das
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Entity-Relationship Model/2

Welcome to module 14 of database management systems. In the last module we will started our discussions on the entity relationship model.

(Refer Slide Time: 00:29)



Module Recap

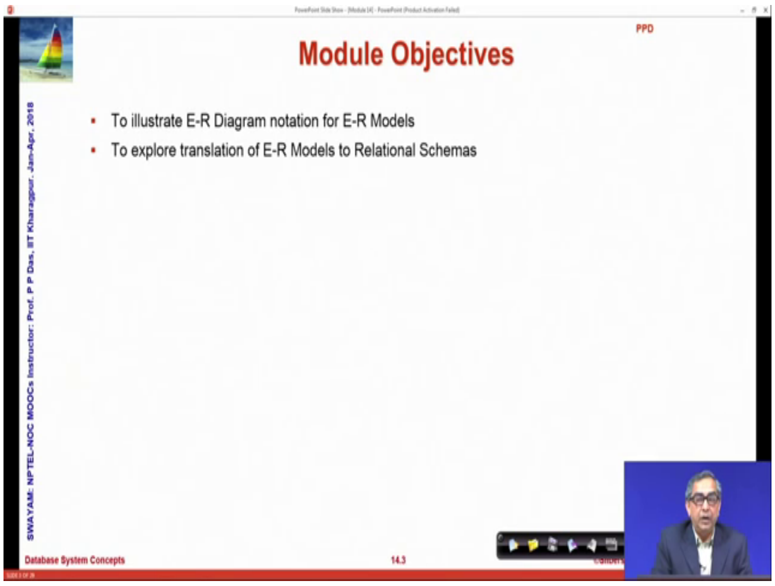
- Design Process
- E-R Model
  - Entity and Entity Set
  - Relationship
    - Cardinality
  - Attributes
  - Weak Entity Sets

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.2 ©Silberschatz, Korth and Sudarshan

We will continue that in this module as well, and actually conclude it in the next module. So, these are the items that we had discussed in the last module.

(Refer Slide Time: 00:39)


Module Objectives

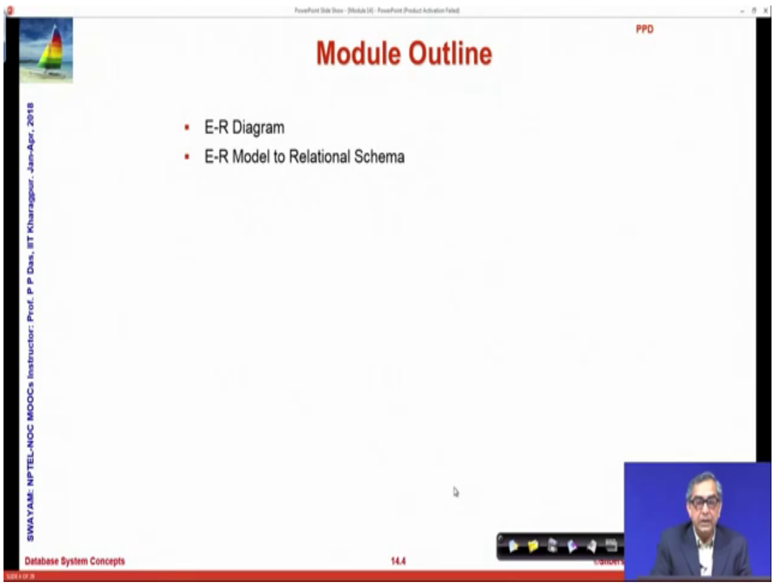
- To illustrate E-R Diagram notation for E-R Models
- To explore translation of E-R Models to Relational Schemas

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.3

In the present one, we will first illustrate the entity relationship diagram notation; that graphical notation for E-R model, how nicely this can be shown in terms of certain diagrams. And then we will explore how E-R models can be translated to relational schemas, which is a basic step of the logical design.

(Refer Slide Time: 01:11)


Module Outline

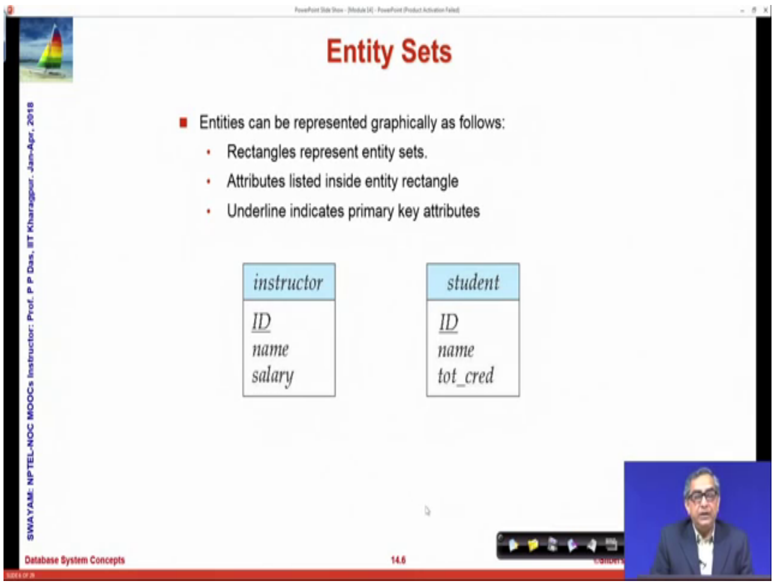
- E-R Diagram
- E-R Model to Relational Schema

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.4

So, these are the topics. So, we start with the E-R diagram.

(Refer Slide Time: 01:16)

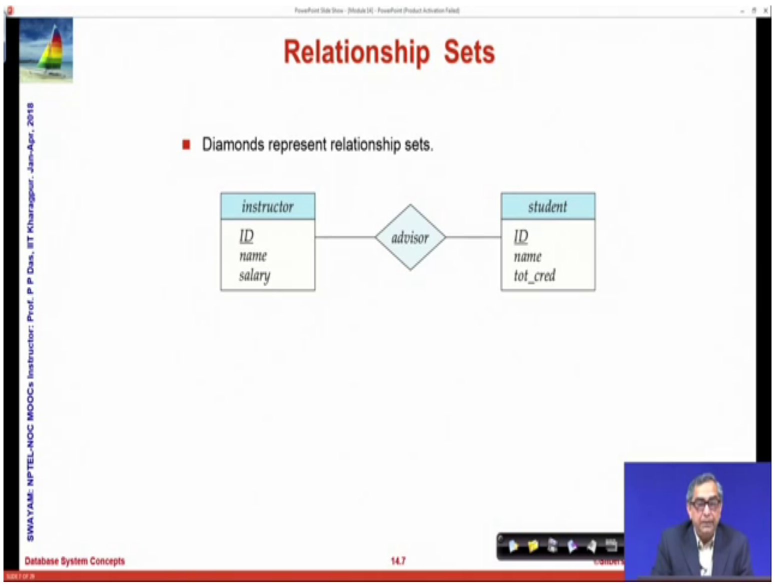


Entity Sets

- Entities can be represented graphically as follows:
  - Rectangles represent entity sets.
  - Attributes listed inside entity rectangle
  - Underline indicates primary key attributes

Naturally the first thing to represent in an E-R model is the entity set. Every entity set is represented by a rectangle. On the top, we write the name of that entity set ss you can see examples here the instructor. And student are the 2 entity sets and below that we write the names of the attributes that are involved. And we underline the attribute or attributes that form the primary key of that entity set.

(Refer Slide Time: 01:49)

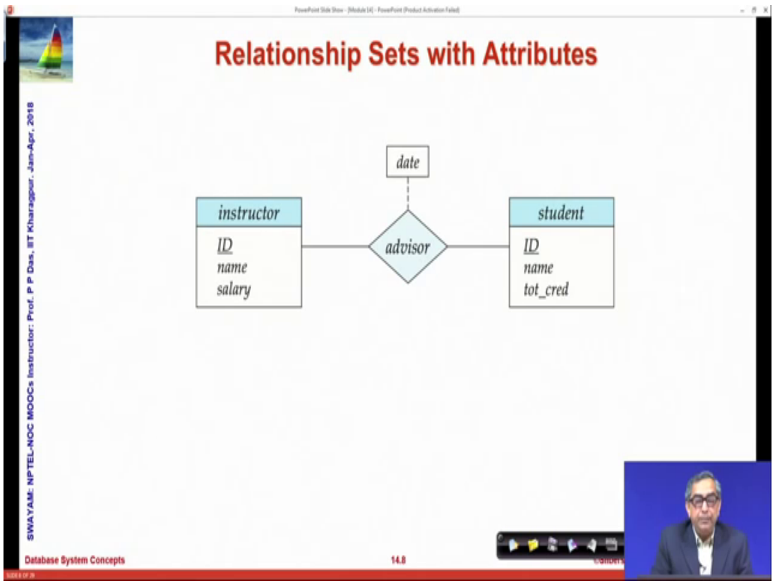


Relationship Sets

- Diamonds represent relationship sets.

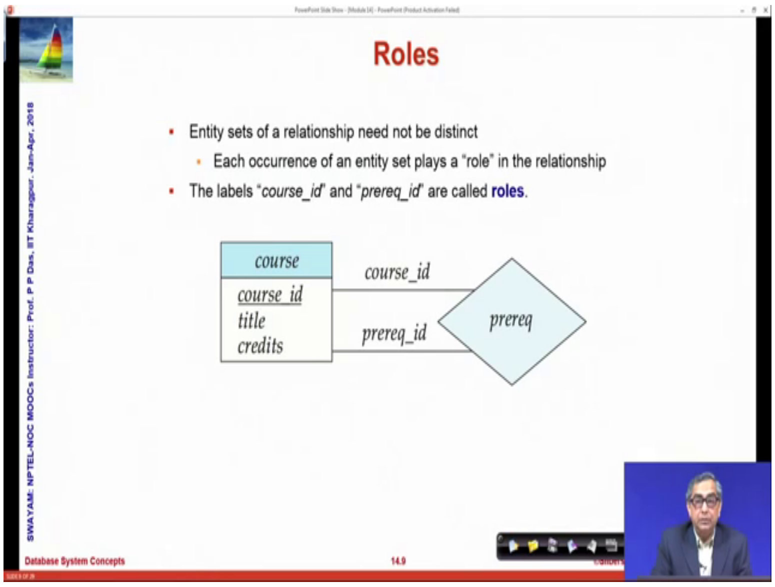
A relationship between 2 entity sets is represented by a diamond, and 2 connecting lines to the 2 entity sets. So, here it says that advisor is a relationship between entity set instructor, and entity set student. Trying to convey the real-world situation that students are advised by the instructors or students have instructors and so on.

(Refer Slide Time: 02:23)


Relationship Sets with Attributes

As we had mentioned that relationships could also have attributes. So, if the advisor relationship has an attribute date then it will be tagged to the adviser relationship with the attribute coming as a within a rectangle and attached to the name of the relationship by a dotted line. So, this shows that advisor is the relationship between instructor. And student and the adviser relationship has an attribute date.

(Refer Slide Time: 02:58)


Roles

- Entity sets of a relationship need not be distinct
  - Each occurrence of an entity set plays a "role" in the relationship
- The labels "*course_id*" and "*prereq_id*" are called **roles**.

course
course_id
title
credits
course_id
prereq_id
prereq
SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018
Database System Concepts
14.9

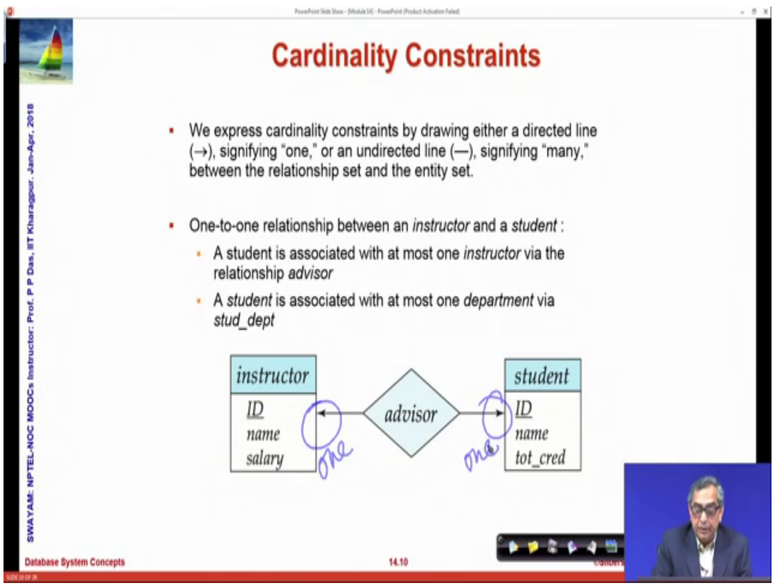
It is possible that the relationship that hold between 2 entity sets can be can use entity sets which are same. That is, it is possible that a set is related to itself.

So, as an example we show, the entity set course which has a relationship prereq, prerequisite which takes a course id, and relates it to another course id called the prerequisite id. Because obviously, if a course has a prerequisite. Then that prerequisite itself is another course id which must occur in this table itself.

So, in this case, if you can see that unlike the earlier case the prereq id is not actually a field of this relation course. So, we say these are rules. So, we say the rule that prereq relate from the course relation to itself are course id and prereq id; where in the actual table both of them relate to course id, but prereq will pair them to show which course has what prerequisite. And we often need this kind of; we have seen similar instances of this while we treated dealt with the recursive queries in databases.

The source destination of aligns problem that we discussed has similar kind of relationship structure. So, you can think about a relationship flies from the set of source to this set of destination, and basically this; these 2 sets the places source places in the destination place are necessarily the same set the same relation. There could be a constraint on the cardinality.

(Refer Slide Time: 05:07)



Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.
- One-to-one relationship between an *instructor* and a *student* :
  - A student is associated with at most one *instructor* via the relationship *advisor*
  - A *student* is associated with at most one *department* via *stud_dept*

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts

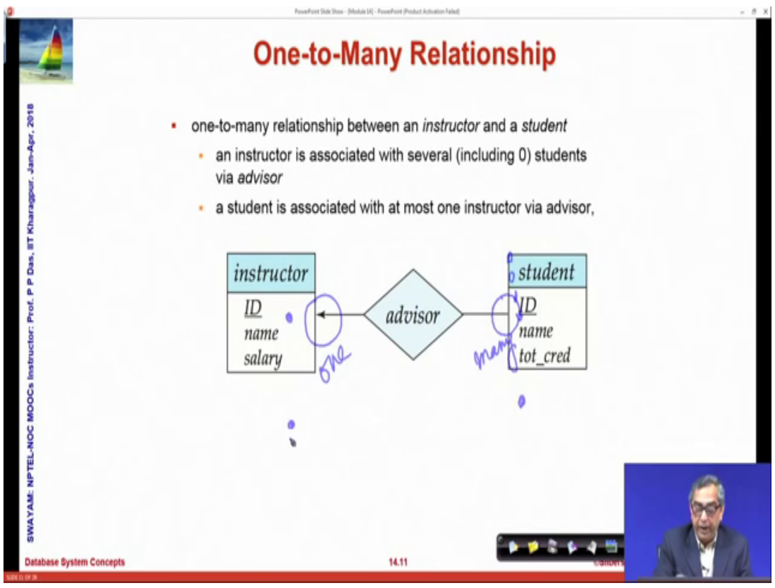
14.10

So, the line that links the relationship diamond with the rectangles of the relations rectangles of the entity sets. Those lines could have an arrow at the end or may not have an arrow at the end. So, if it has got an arrow, then it means that suppose it has an arrow it means 1. And if it does not have a arrow if it is simple then it means many. So, using this rotation we can designate one to one to many these all different kinds of cardinalities that we had discussed.

So, for example, if we are showing this arrow on both hands, both ends of this relationship advisor then, it means that it is a one to one relationship because there is an arrow here. So, this is 1. There is an arrow here. So, this is 1. There is an arrow here. So, this is 1; which means that a student is associated at most with at most one instructor. And it also means that and an instructor is associated with at most student.

This may not be a reality this usually is not the reality, but this we are just showing this as an example. So, if the student instructor relationship advisor relationship is one to one, then this is how we will denote it.

(Refer Slide Time: 06:54)



One-to-Many Relationship

- one-to-many relationship between an *instructor* and a *student*
  - an instructor is associated with several (including 0) students via *advisor*
  - a student is associated with at most one instructor via advisor,

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

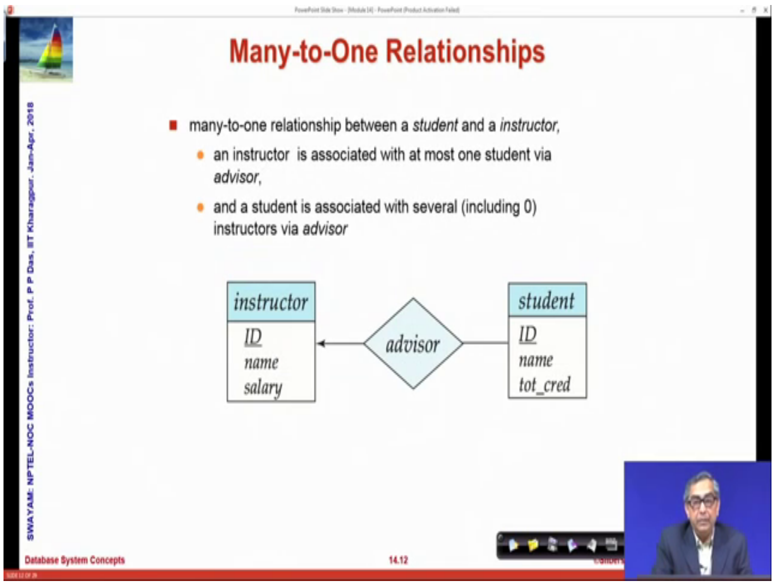
Database System Concepts

14.11

If it is one to many; so, which side is 1, this side is 1 and this side is many. So, it is one too many from instructor to student; which says that every student has at most one instructor. And an instructor may have several students. It could be null also it could be none also.

So, for an instructor here there are many students, but for a student here there are only at most one instructor. So, it is one to many and this is how we designate.

(Refer Slide Time: 07:35)



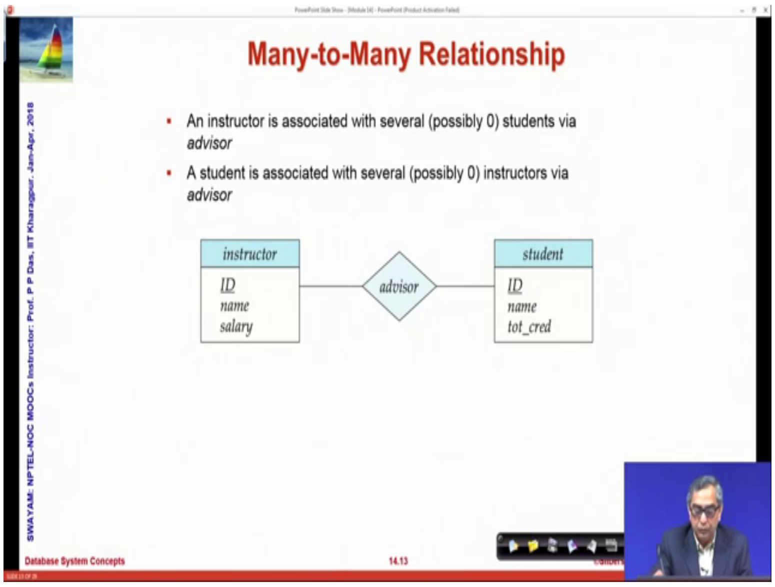
Many-to-One Relationships

- many-to-one relationship between a *student* and a *instructor*,
  - an instructor is associated with at most one student via *advisor*,
  - and a student is associated with several (including 0) instructors via *advisor*

A similar thing will happen, if I read the same relation in the other direction. So, instructor to student was one to many. So, student instructor also drawn in the same way, because this is this is the one side, this is the one side and this is the many side.

So, if the situation is the same. So, if we read from the student instructor, then it is also designates the many to one relationship.

(Refer Slide Time: 08:05)



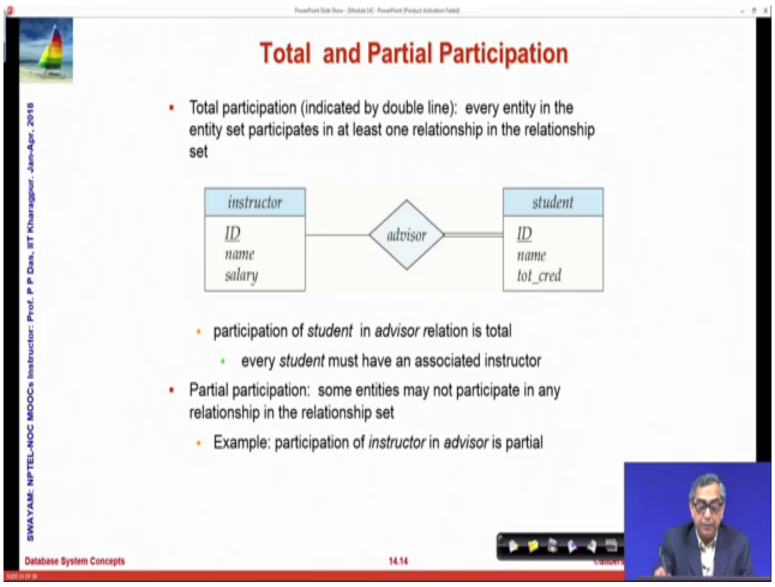
Many-to-Many Relationship

- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*

And finally, we can have a many to many relationship, where there is no arrow at either end which means that an instructor is associated with; several possibly none no student why the advisor relation and the student may also have several instructors by the advisor relationship.

Now, we can you can certainly figure out that in the particular case of student instructor scenario of providing advice, one to one as well as many to many are not the usual real world scenarios, but these are we have just using to show you how to model this usual scenario would be from instructor to student it is one to many relationship.

(Refer Slide Time: 08:50)



Total and Partial Participation

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set

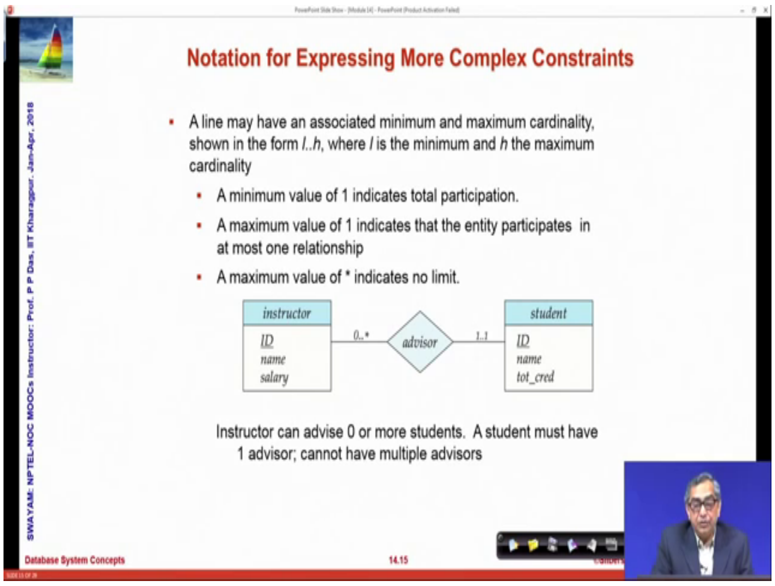
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018
Database System Concepts
14.14

- participation of *student* in *advisor* relation is total
  - every *student* must have an associated instructor
- Partial participation: some entities may not participate in any relationship in the relationship set
  - Example: participation of *instructor* in *advisor* is partial

A relationship could be total or it could be partial. If you if one side of the relationship, or whichever side of the relationship is total, then we draw a double line. So, you can you can see here in the diagram we are drawing a double line, which means that in the advisor relationship the involvement of the student is total, which means that every student must feature in the advisor relationship. Or in other words every student must have an advisor. But it is partial on the instructor side because every instructor may not have a student.

So, this double life shows that reality. Some entities may not participate in any relationship is a partial.

(Refer Slide Time: 09:59)



Notation for Expressing More Complex Constraints

- A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality
  - A minimum value of 1 indicates total participation.
  - A maximum value of 1 indicates that the entity participates in at most one relationship
  - A maximum value of * indicates no limit.

Instructor can advise 0 or more students. A student must have 1 advisor; cannot have multiple advisors

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018
Database System Concepts
14.15

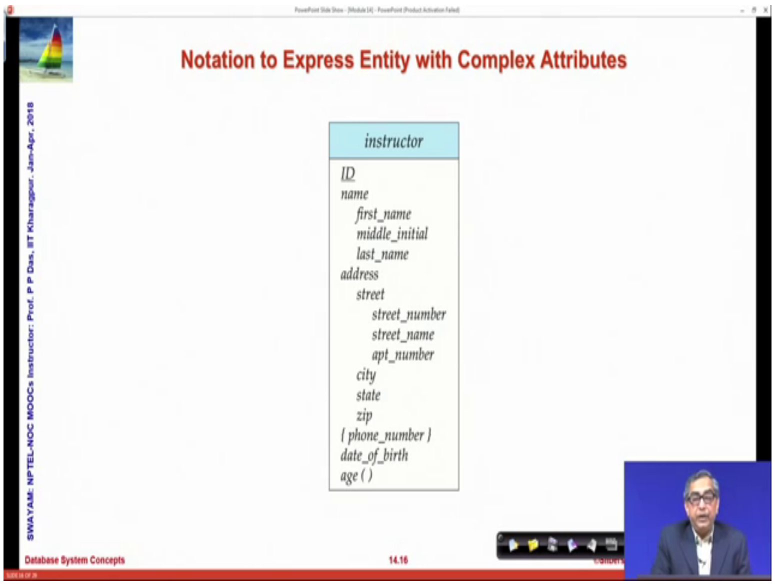
Now, these constraints the cardinality constraints can be made more precise by actually using numbers. You can actually say on the 2 sides of the relationship, that at the minimum how many entities should relate, and at the maximum how many entities can relate. For example, if we are saying that we are on the right-hand side here, if you see we are saying that it is maximum minimum is one maximum is one which what does it say it says that every student the minimum is one. So, every student must feature in the advisor relationship.

So, in real world, every student must have a an advisor, must have an instructor. It says maximum is one; which says that every student can have at most one instructor. So, this one to one one, dot dot one says that every student must have at least one instructor, every student must have at most one instructor. So, together it says that every student must have exactly one instructor.

Whereas if I if we see on this side, it says that 0 dot, dot, star, star stands for no limit. It can be anything any number. So, the minimum is 0 which means that an instructor may not have a student. And star says that if the instructor can have any number of student. Naturally 0, 1, 2, 3, 4, 2, or 200. So, any instructor can advise any number of students.

So, these kind of precise number constraints can be put in addition to the one to many, or one to one, many to many kind of notations in the diagram. So, when we do that we have the precise cardinality of the complex relations that exist.

(Refer Slide Time: 12:10)



Notation to Express Entity with Complex Attributes

instructor

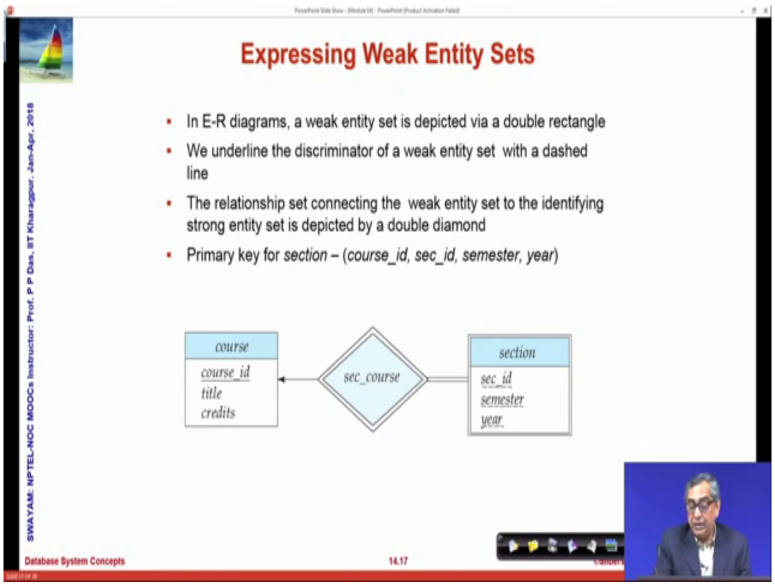
ID
name
  first_name
  middle_initial
  last_name
address
  street
    street_number
    street_name
    apt_number
  city
  state
  zip
{ phone_number }
date_of_birth
age ( )

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.16

Next, we take a look into the handling of the complex attributes. The first you remember that, the first kind of complex attribute is 1 which is composite. Say, name which has first name middle name, initial middle initials and last name.

(Refer Slide Time: 12:26)



Expressing Weak Entity Sets

- In E-R diagrams, a weak entity set is depicted via a double rectangle
- We underline the discriminator of a weak entity set with a dashed line
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond
- Primary key for *section* – (*course_id, sec_id, semester, year*)

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.17

So, when we have that, then the way we represent is at the actual name of the attribute is at the outermost level. And it is composites are written with certain shift on the left. So, these all say that these are composites of name. So, if this says that street city state zip are composites of address, and further indentation say, that these are composites of street. So, this is how graphically we show that how complex attributes feature.

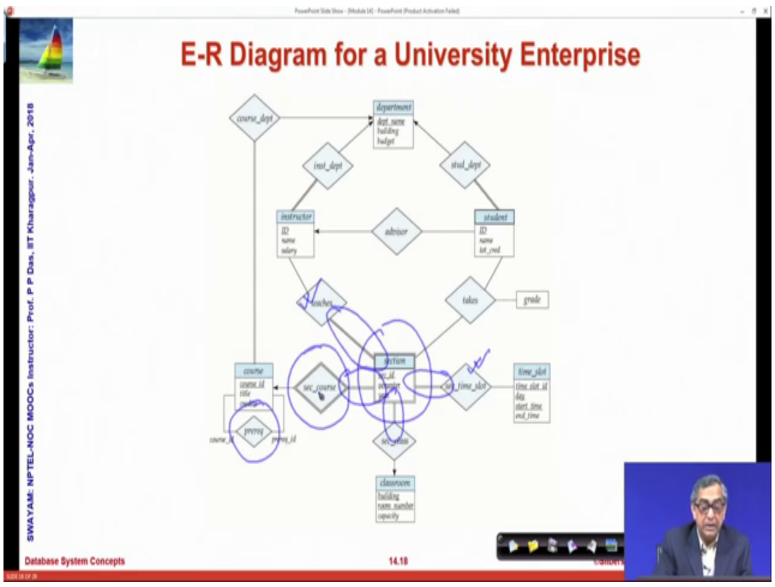
Now, let us go back to discussing the weak entity sets. In the E-R diagram, a weak entity set is represented by a double rectangle. You remember the section is a weak entity set. And why is it so? Because the same course may have 2 different, I am sorry, 2 different courses may have the same section id semester and year that is 2 courses 2 or more courses may run sections by the same name in the same semester and the year. So, a section cannot be uniquely identified by these 3 attributes. It needs a relationship with the identifying entity set course to be specific the course id so that the entities here in can be uniquely specified.

So, since this has happened. So, we designate that by putting this double rectangle around the weak entity set section. We underlined the discriminator of a weak entity set with dashed line. So, you remember these are the discriminators. Because given the identifying attribute in the identifying set. These are the attributes which distinguish different tuples of section. So, they are not shown with solid underline, they are shown as dotted underline, dashed underlined. So, that you can make out that this is a weak entity set and these are the discriminators.

The relationship set connecting the weak entity set to the identifying strong entity set, is also so, this the moment you have weak entity set. You know that there has to be a relationship to the strong entity set which identifies it. So, that relationship set course which say, course id against this minds that is designated with a double diamond. So, that you know that this is the identifying relationship between a weak entity set and the corresponding strong entity set. And once that happens in the primary key becomes the discriminators of section the weak entity set, and the primary key of the identifying a strong entity set the course. So, that forms our final primary key for this entity set section.

My new course id is not a part of this relation, but it actually plays the role through this section id as a key for the section relation without which the section entities in the section cannot be uniquely identified.

(Refer Slide Time: 16:28)



E-R Diagram for a University Enterprise

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.18

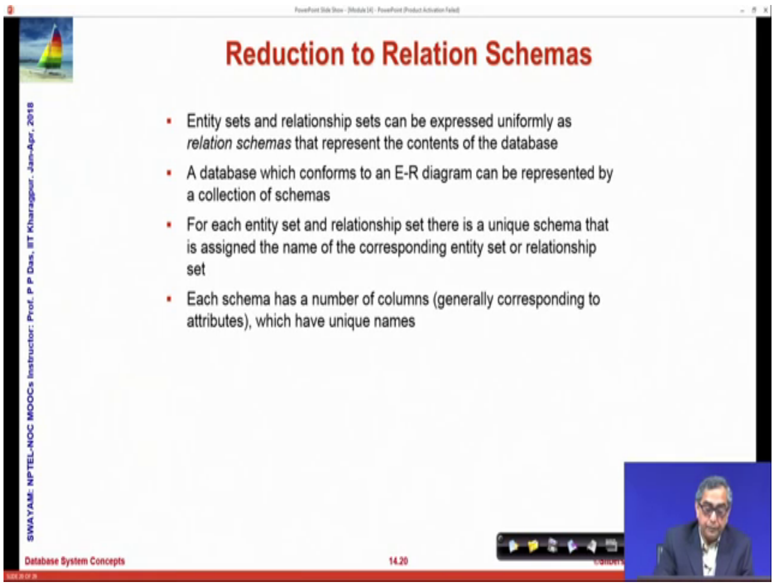
So, having said that this is a the E-R diagram of the university enterprise. Some of the points that you could take a look at; this is the weak entity set we have just seen. This is the relationship to the identifying strong entity set. This is a prerequisite multi role relationship. this we can see is a is a total involvement. So, worry why is it a total involvement, because every section must have at least one teacher. So, there cannot be a section which does not feature in the teacher's relationship. Similarly, every section must get a timeslot where the classes for that section is held.

So, every section must feature in the sec timeslot. So, these are the similarly, it must get a classroom. So, these are all different total involvements. That we total roles that we can see we can see some of that elsewhere as well. For example, you can see it here, we can see it here, because in between instructor and the department the inst department relationship. Certainly, every instructor must have a department. So, it is total, but it is not the same for the department, every department will not have instructors.

So, these is how if we can you can go through carefully. And for example, this is another which is total, which means that every course need a department you cannot run a course which does not have a department. So, this is how we can see that how the E-R diagram that the first conceptual level diagram of a very simple university enterprise is being designed following the notions and symbols of E-R model that we have already developed.

Next comes the part where, from this model which is primarily diagram based we have to really go to the relational schema, which is names of relations and attributes which is pretty much a straightforward job.

(Refer Slide Time: 18:41)



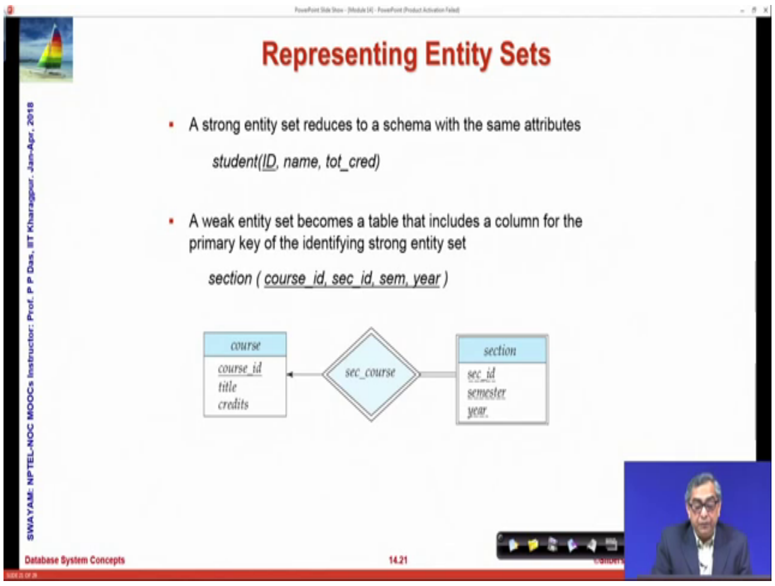
Reduction to Relation Schemas

- Entity sets and relationship sets can be expressed uniformly as *relation schemas* that represent the contents of the database
- A database which conforms to an E-R diagram can be represented by a collection of schemas
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set
- Each schema has a number of columns (generally corresponding to attributes), which have unique names

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018
Database System Concepts
14.20

So, entity sets and relationship sets have to be represented in terms of relational schema. What is the beauty of the E-R model? And the relational schema is that that, when you reduce the entity relationship model to relational schema both entity sets and relationships sets. Both of them turn out to be relational schemas. And so, that the database finally, can be represented simply as a set of schemas each one of which must have a set of identifying primary key.

(Refer Slide Time: 19:20)



Representing Entity Sets

- A strong entity set reduces to a schema with the same attributes

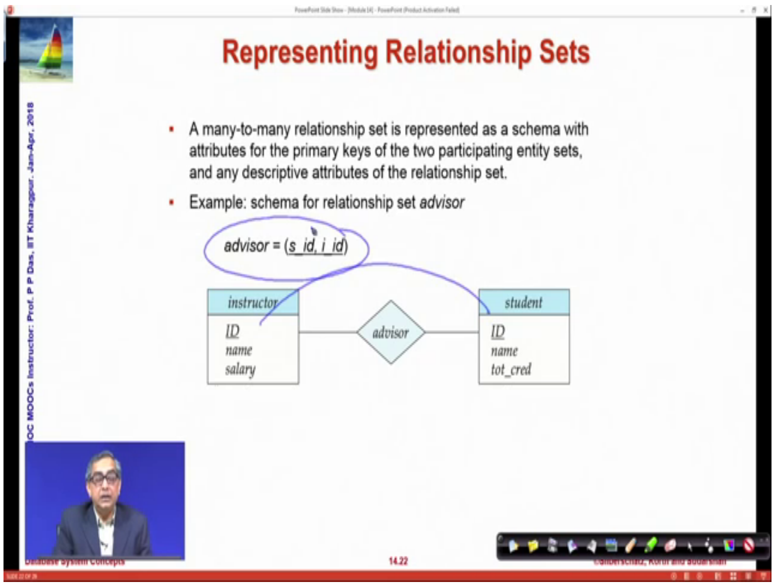
  student(ID, name, tot_cred)

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

  section (course_id, sec_id, sem, year)

So, let us look into that. So, on the strong entity set, that reduces to schema with the same attribute as student. So, student has id name and total credit so, which we saw earlier. Now this gets converted to a schema with id being the primary key. The other case of weak entity set, section which had the 3 discriminators, and through set course relationship was identified from the strong entity set. Course borrows the primary key of the course to be defined in terms of this relational schema.

(Refer Slide Time: 20:05)



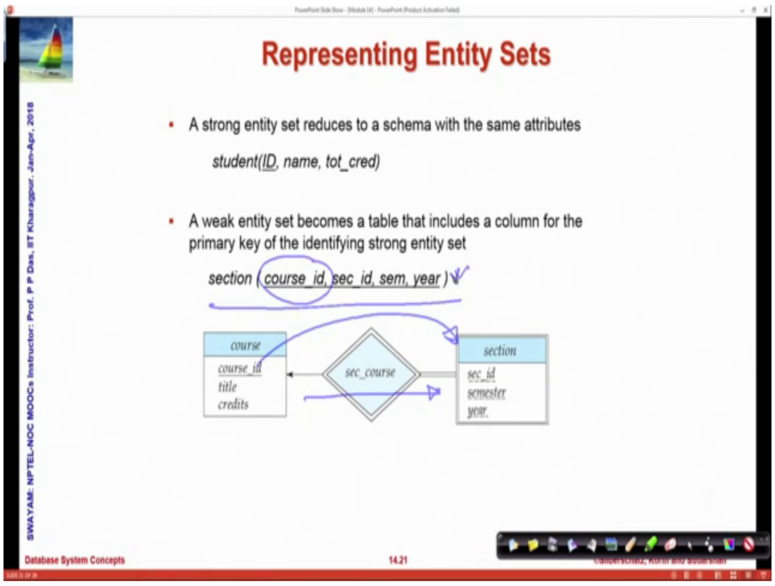
Representing Relationship Sets

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set *advisor*

  advisor = (s_id, i_id)

One moment; this borrows the primary key from here and becomes.

(Refer Slide Time: 20:10)



Representing Entity Sets

- A strong entity set reduces to a schema with the same attributes

  student(ID, name, tot_cred)

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

  section (course_id, sec_id, sem, year)

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

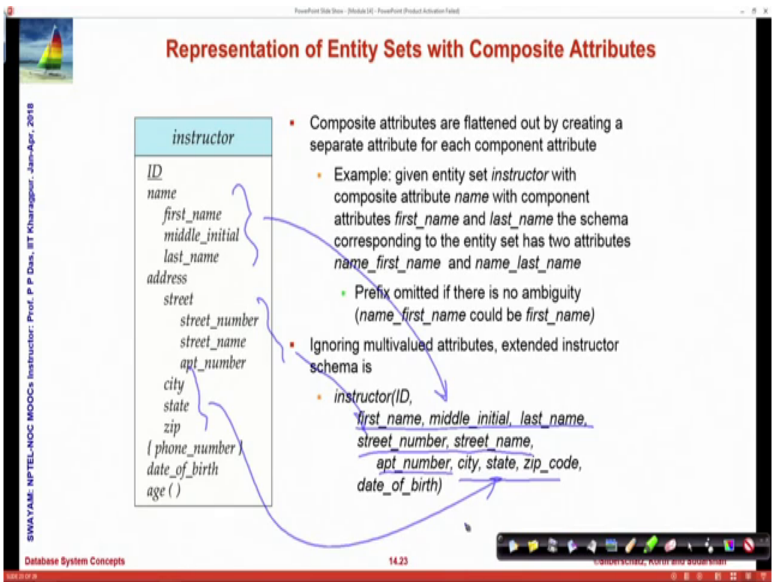
Database System Concepts 14.21 ©Silberschatz, Korth and Sudarshan

So, you can see that in the E-R model, the section did not have course id as an attribute, but while we reduce this to the relational schema through this sec course relationship, we have borrowed this primary key from course. The primary key of course, the course id and added that to section to make it a complete relational schema.

Next comes the representation of relationships. So, we are showing a relationship advisor so, which relates instructors to students. So, naturally every instructor is identified by id every student is identified by id. Since both of the attributes have the same name id. We are calling them as s underscore id and for the student and I underscore id for the instructor. So, the advisor relation is basically a pairing of these 2 ids which gives rise to a relationship which looks like this relationship schema which looks like this. So, it can in general say that if we have a relationship in the E-R model, which we want to represent in the schema then we will take, we will create a schema which has the primary key of both the sets, and put them together. And if the names clash we will just change the name with the name of the relation, and that will give us the schema for the relationship in this case the advisor.

So, you have seen how to represent entity sets, weak entity sets and relationships.

(Refer Slide Time: 22:03)



Representation of Entity Sets with Composite Attributes

instructor
ID name first_name middle_initial last_name address street street_number street_name apt_number city state zip { phone_number } date_of_birth age ( )

- Composite attributes are flattened out by creating a separate attribute for each component attribute
  - Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*
    - Prefix omitted if there is no ambiguity (*name_first_name* could be *first_name*)
- Ignoring multivalued attributes, extended instructor schema is
  - *instructor(ID,*
    first_name, middle_initial, last_name,
    street_number, street_name,
    apt_number, city, state, zip_code,
    date_of_birth)

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

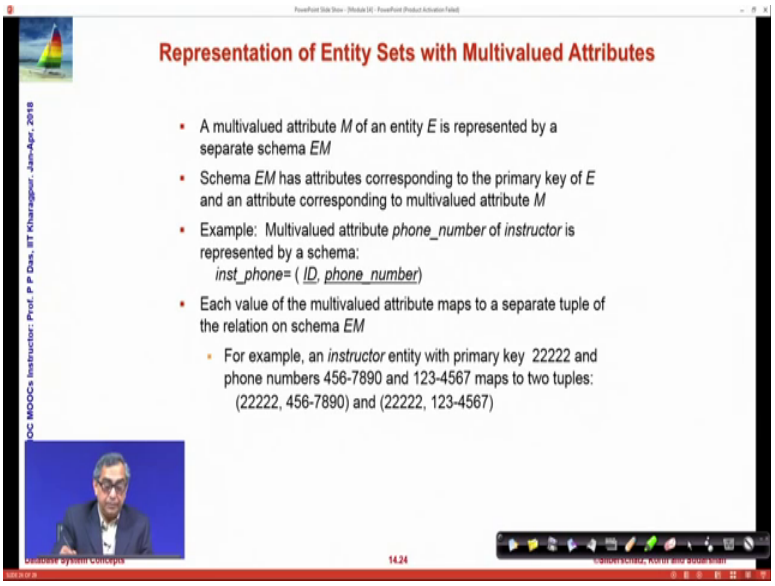
Database System Concepts 14.23 ©Silberschatz, Korth and Sudarshan

Ah let us look at how do we deal with composite attributes. Because so far, we had assumed that the relational schema has attributes, and every attribute has a domain. The type from where it is values come. So, if I have a composite attribute, where every attribute has a set of components. Then the easiest way to handle this is to what is known as flatten the composite attribute.

So, flattening basically is for example, if I take a name it has 3 components. So, each one of them I can call by given new name, name underscore first name, name underscore middle initial name underscore last name. By prefixing with the attribute name, I make the names of these components necessarily unique. Now after I do the prefix thing I might figure out that actually prefixing is not required first name itself is a unique. Because it does not occur anywhere else if it is then I can I may drop the prefix name. But in general, I can take the attribute name prefix on the component, and just flatten them out make them all attributes each separate attribute.

So, here when we flatten out we will have first name, middle, initial, last name as you can see these flattened out from here. Then we have street number, street name, apartment number, flattened out from the street. Subsequently, we have city state zip flattened out from here. So, all of them flattened out has become separate attributes. And flattening is a very straightforward mechanism by which you can convert complex composite attributes into the regular schema design getting to little bit of issue if you have a multi valued attribute.

(Refer Slide Time: 24:02)



Representation of Entity Sets with Multivalued Attributes

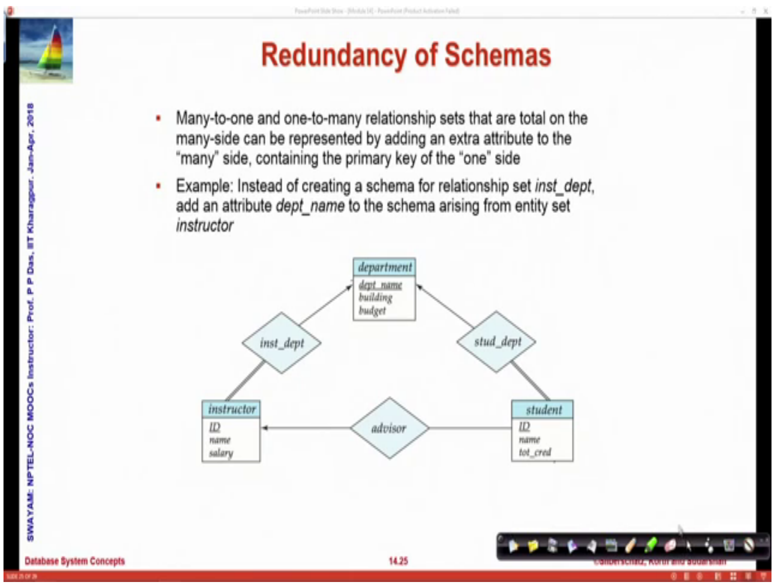
- A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*
- Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*
- Example: Multivalued attribute *phone_number* of *instructor* is represented by a schema:
  inst_phone= (ID, phone_number)
- Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*
  - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples: (22222, 456-7890) and (22222, 123-4567)

Multivalued attribute is one where one attribute may have multiple values at the same time. And the example we talked about is a phone number.

I may have multiple phone numbers. So, certainly against an attribute I can keep only one value. So, if I if my attribute is multiple value. Then the basic idea is to use a separate schema to maintain this multiple values. For example, if I have to maintain multiple phone numbers of an instructor. I may have a separate I may decide to have a separate relation which relates the key of the instructor relation, and the attribute that I want to maintain multipally.

So, in this relationship in this relation inst underscore phone against the same id I can have different phone numbers. So, there will be different records which match on the id, but do not match on the phone number which gives me the different values that the phone number can take. And then this inst phone in conjunction with the instructor relation will actually denote the multi valued phone number attribute. So, this is just an example showing that for one primary key of an instructor 1 to 2 2 2 2 and there are 2 phone numbers. So, this will basically mean you have 2 tuples in the new relation.

(Refer Slide Time: 25:37)



Redundancy of Schemas

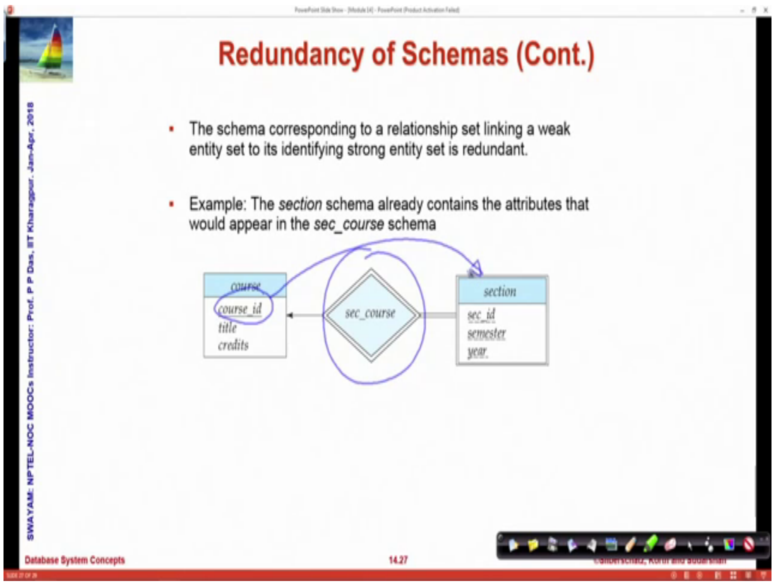
- Many-to-one and one-to-many relationship sets that are total on the many-side can be represented by adding an extra attribute to the "many" side, containing the primary key of the "one" side
- Example: Instead of creating a schema for relationship set *inst_dept*, add an attribute *dept_name* to the schema arising from entity set *instructor*

So, with that we can handle multiple multivalued attributes also. Some of the relationships that we may have modeled, which we have done in doing the database E-R schema could have redundancy. For example, take a case here we have the instructor. And we have the student advisor relation is incidental here, and we have department. So, we want to say that the instructors belong to certain departments. Every instructor belongs to one department which is the totality of the relationship here.

Similarly, every student belongs to a department, totality of the relationship on this side. And inst dept inst dept in that context is a relationship which is between instructor and department. Similarly , so, we can there is certain redundancy in this, because we can get make this simpler if we just take the primary key of this relation and put in here. If we do that then basically this become redundant these are no more required. So, all that you are saying is the instructor has a dept name field which says which department does it belong to.

So, if there is a choice between whether you will keep such relationships or you will actually reduce the redundancy in the schema, and involve the primary key of the other relation into your primary table which is instructor or the student here. So, instead of creating a schema for relationship set inst department. You will simply add a department name. So, mind you this is at the at the E-R model level you did have a separate relationship, but while you reduce it to your relationship relational schema you are reducing that relationship by including the dept name as an attribute in instructor.

(Refer Slide Time: 27:53)



Redundancy of Schemas (Cont.)

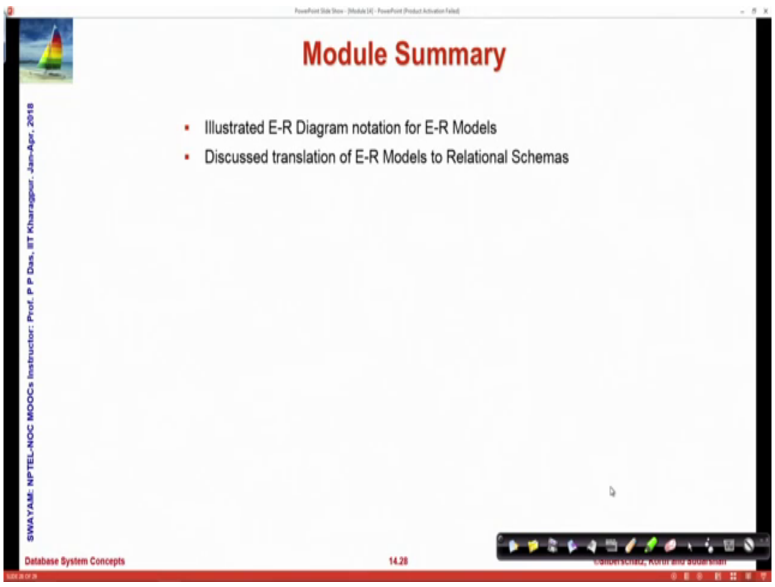
- The schema corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant.
- Example: The *section* schema already contains the attributes that would appear in the *sec_course* schema

course
course_id
title
credits
sec_course
section
sec_id
semester
year
SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018
Database System Concepts
14.27
©Silberschatz, Korth and Sudarshan

So, that is called the reduction of schema which is often used. for one to one relationship. So, this is this, was this is good if you have many to one relation. Because this was possible, because every instructor has one department. So, if you just include the department name with the instructor. Or every student has one department ah. So, it is possible that way, but if you have a one to one relationship, then naturally you can do the similar reduction by I by choosing the either side as the many. You know, because because the the unique side has to come on the many side. The unique side here. This is the unique side here. Because every instructor has a unique department and this is the many side here.

So, the unique side has to attribute has to come in here. The unique side primary key has to come in here. So, instead of so, you can apply the same principle to a one to one relationship by treating any one of them as a many side, and add the extra attribute on the other side to get rid of this additional schema. The schema corresponding to a relationship set linking a weak entity set to it is underlying strong entity set is certainly redundant, we have already seen this. So, this (Refer Time: 29:13) is made redundant by including the primary key of the identifying relation, identifying entity set in the weak entity set. So, that is another reduction of schema that can be done.

(Refer Slide Time: 29:31)



Module Summary

- Illustrated E-R Diagram notation for E-R Models
- Discussed translation of E-R Models to Relational Schemas

SWAYAM: NPTEL-NOC MOOCs Instructor: Prof. P P Das, IIT Kharagpur. Jan-Apr, 2018

Database System Concepts 14.28 ©Silberschatz, Korth and Sudarshan

So, to summarize we have in this module illustrated the entity relationship diagram, which are very nice ways of graphically representing what we see in the real world. So, it has graphical representation of entity sets, attributes the key attributes primary key attributes the weak entity sets, and the relationships along with the cardinality information.

And then we have shown that using certain reduction rules how we can easily reduce this entity relationship diagram or entity relationship model into the traditional relational schema. And we have seen that both the entity sets as well as the relationship sets become relational schemas.