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

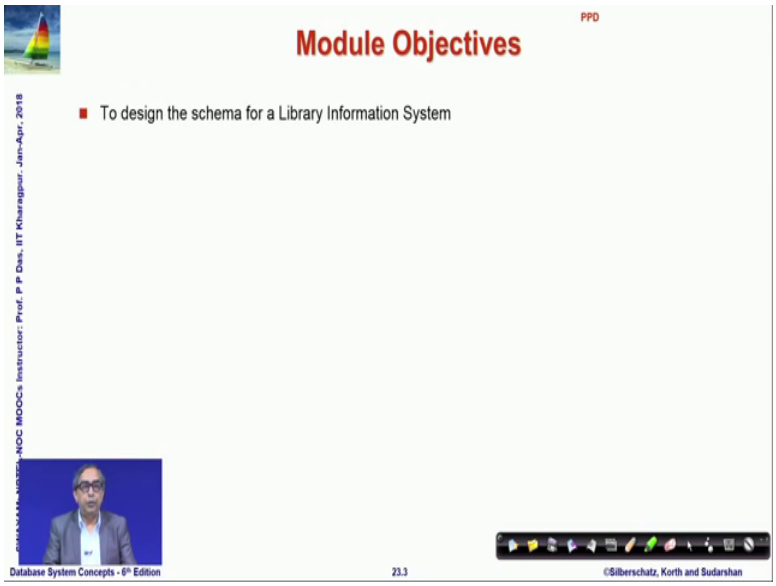
Lecture – 23
Application Design and Development (Contd.)

Welcome to module 23 of Database Management Systems. We have been discussing about application design and development and this is the third and concluding module in that regard.

(Refer Slide Time: 00:27)

In the last module we have discussed about different aspects of application architecture rapid development process issues and performance and security and took a glimpse in terms of, what is required for doing a mobile app.

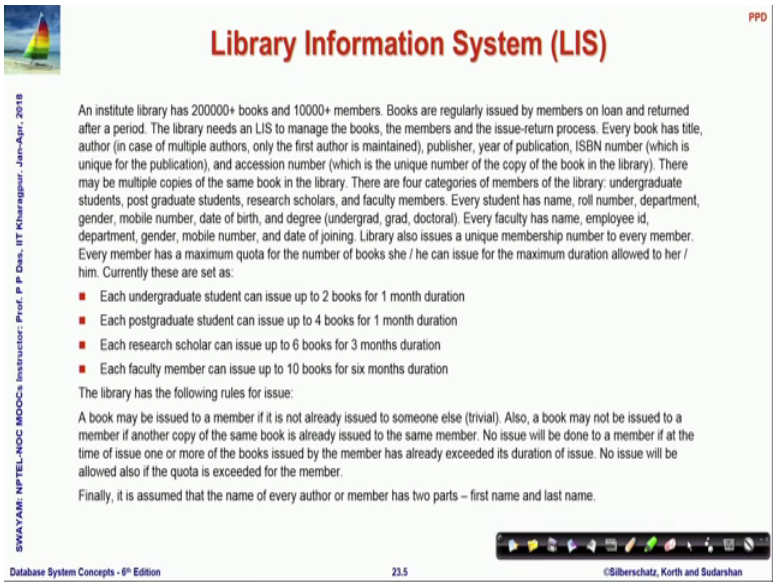
(Refer Slide Time: 00:44)



In this module, we will take care case study in terms of a library information system. We will try to design the schema for that as you have seen that for a database application design there are frontend designs there are middle tier business logic design and there will be issues in terms of the database design.

Since we are in the DBMS course, I will not focus on the whole aspects of application design, but we will focus specifically on the database aspect. So, we will start with a basic requirement specification for a library information system and then from the starting from that we will try to extract different entities and attributes and their relationships and we will make a relational schema and use different notions of dependency and how to write queries we will look into those aspects and refine that and finalize a schema for this problem. So, let us work through that. So, the outline the module issue is library information system.

(Refer Slide Time: 01:55)



So, let us start with a basic this is a very small description. So,, but yet it will give us lot of food for thought in this work and while you actually go through the rest of the video. I would suggest that you take a print out of this page or keep this page separately because you will frequently need to refer to it.

So, let me quickly go through this we are talking about an institute library that has over 2 lakh books and over 10,000 members who can use regularly issue the books on loan and return them on the expiry of the period or before that and the library needs a library information system to manage the books the members and as well as the issue return process. So, we are just looking into these aspects not the procurement of books and organization of the books and so on.

Now what is given every book has it is title author publisher etcetera a number of different attributes ISBN number accession number in terms of a book I must explain to you at this stage that any book that you that is published has an ISBN number which is an international number given so, that you can uniquely identify that book.

So, if we are talking about the database management book we are following here then that has a ISBN number, but that does not mean that number actually is unique for the book would not for a specific copy of the book. So, if you buy three copies of the book and put it in the library, then these three copies need to be identify separately by another number which is typically called the accession number.

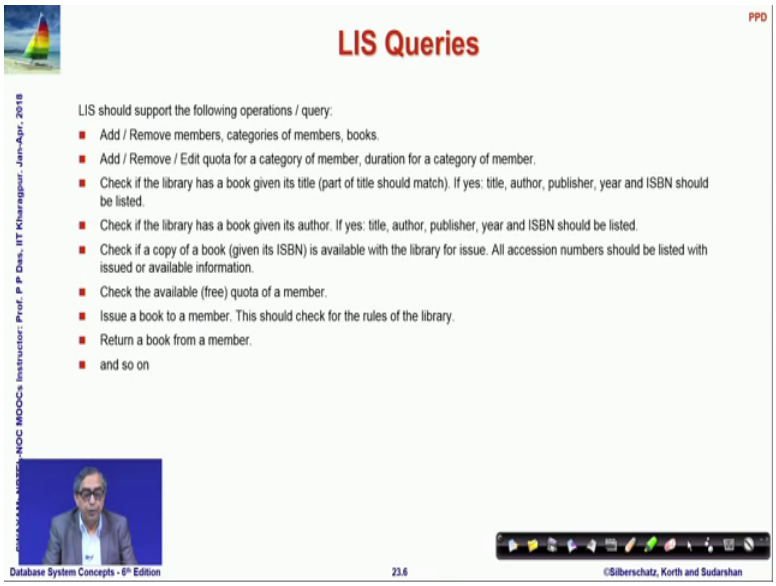
So, naturally the LIS need that every book has ISBN number which says which book it is and the accession number which says which specific copy it is, because they are may be multiple copies of the same book in the library on the member side. There are 4 categories of members broadly: students and teachers, but amongst students if they could be undergraduate postgraduate or research students and the faculty members. The student are specified by their name, roll number, department, gender, mobile number, date of birth, and the degree that they are doing and every faculty member also has a name, employee id, department, gender, mobile number.

And the date of joining the library to manage these members library also issues a unique membership number to every member and every member has a maximum quota for the number of books that she or he can issue and the maximum duration for which those books can be retain once issued. So, there are different specification for at different categories of member can have different quota and different duration the general rule as specified by this libraries the a book may be issues to a member naturally if it is not issued to someone else the book has to be available also a book may not be issued to a member if another copy of the same book is already issued to the same member.

So, you cannot issue two copies of the same book at the same time no issue will be done to a member if he is kind of a default that is the time of at the time of issue one or more of the other books already issued by the member has already exceeded the duration of the issue.

So, they are overdue for return in that case no issue will be allowed no issue will; obviously, be allowed if the quota x x and it is also specified that whenever we talk about name every name will have two parts the first name and the last name. So, this is the this is the given specification based on this we will need to make a good relational schema to represent the tables and manage the queries.

(Refer Slide Time: 05:45)

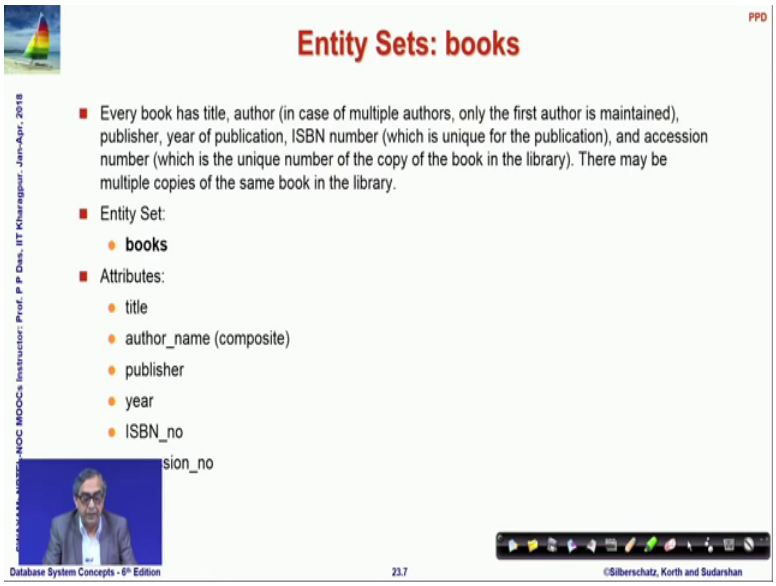


So, let us take a quick look into some of the sample queries this is just a indicative sample. Now, naturally we need a whole lot of you know insert delete update kind of queries for adding or removing members categories of members shapes the books and so, on adding or removing or changing the quota of a category of member the duration for that also we will have queries like to check.

If the library has a given book with a given title and if it is found then the details of those should be listed or we need to check if library has a book given it is author. So, if I say the author it should be possible to locate a book we should able to check, if a copy of a book if I specify the ISBN number, then whether it is available with the library for issue. So, there may be as I said multiple copies of the san book.

So, given the ISBN number it will return all the copies the accession number of all the copies and their issue status; whether they are issued or they are available it should be available it should be possible to check the quota available free quota of a member and certainly it should have features of issuing a book to a member and they should check for the rules of the library as stated it should be possible to return a book and so, on. So, these are the typical queries against which in the backdrop of which we will try to design the database system. So, what we will do?

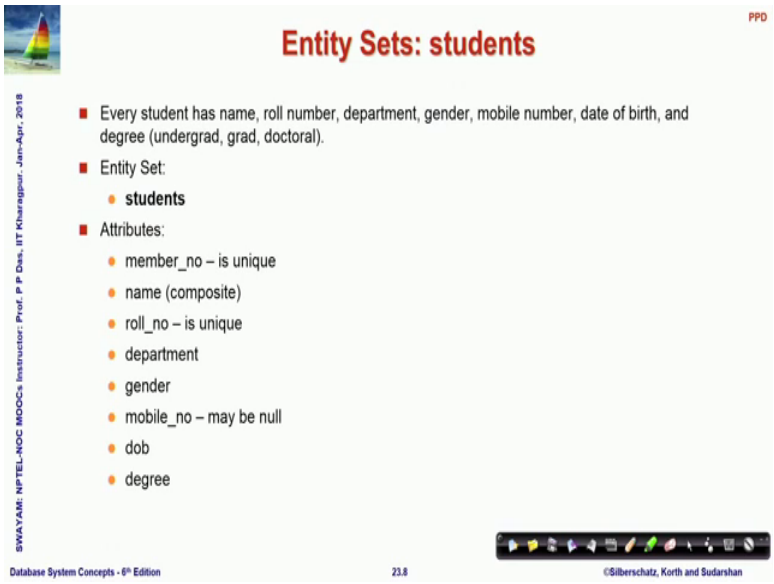
(Refer Slide Time: 07:18)



We will initially start with a specification as given. So, I hope you already have kept a copy to refer to and we will try to extract the different entity sets and the attributes. Naturally, the first entity said that we extract we let us call it books which is about books where in the beginning I have given the basic statement that is given in the so, in the specification.

So, from that we can see that books will be an entity set and it will have a attributes like title author name which is a composite one, because it will have two parts into that the first name and the last name the publisher year ISBN number and accession number. So, these are the attributes available for books. So, this is my first entity set.

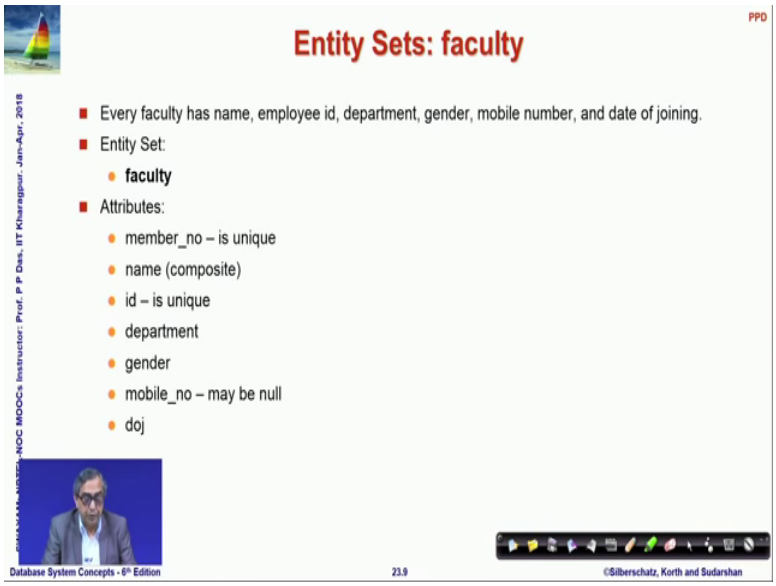
(Refer Slide Time: 08:11)



The second entity set are students. So, this is the statement about the student and from that statement, we can easily extract that entity set here is student and the attributes are member number, because certainly in the context of the library system as we said the; student has to be a member. So, it should be there will be a member number which is has to be unique the composite name the student has a role number. So, we assuming that the role number is a unique field.

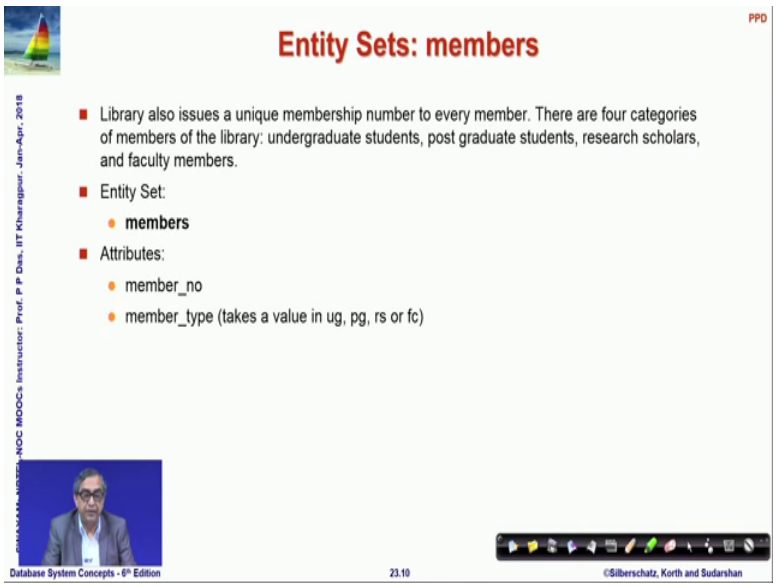
So, knows to students will have the same roll number, then there is department gender mobile number which could be null the; date of birth degree that the student has done is doing. So, those are the different attributes for this entity set.

(Refer Slide Time: 09:03)



So, we have books we have students similarly we will have faculty again the there is this process will continue by extracting different parts from the specification. So, these are statement about the faculty and we know that there will be a faculty entity set with attributes like member number name id and so, on.

(Refer Slide Time: 09:25)

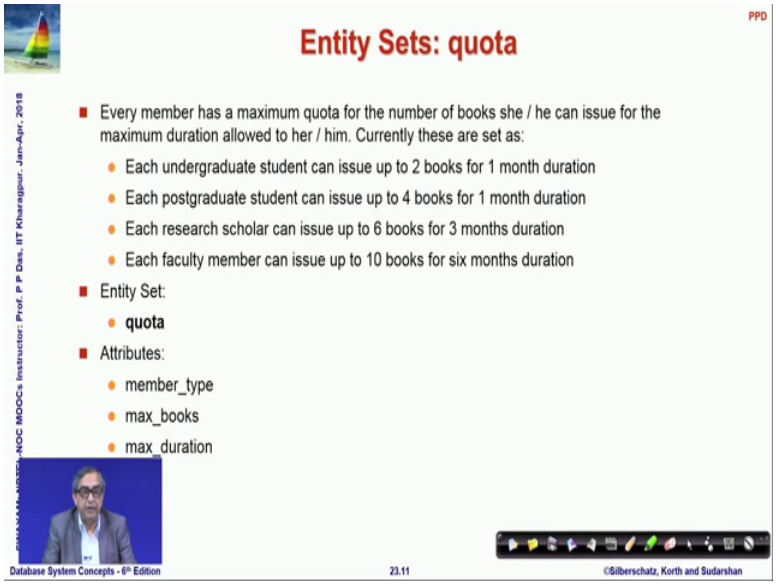


So, it should be quite obvious library has talked of that it can each it will issue unique membership number to every member and there are 4 categories of member. So, let us

we are just making attentive you know suggestion that there could be. So, it looks like members are in entity which the library has to interact with in terms of issue.

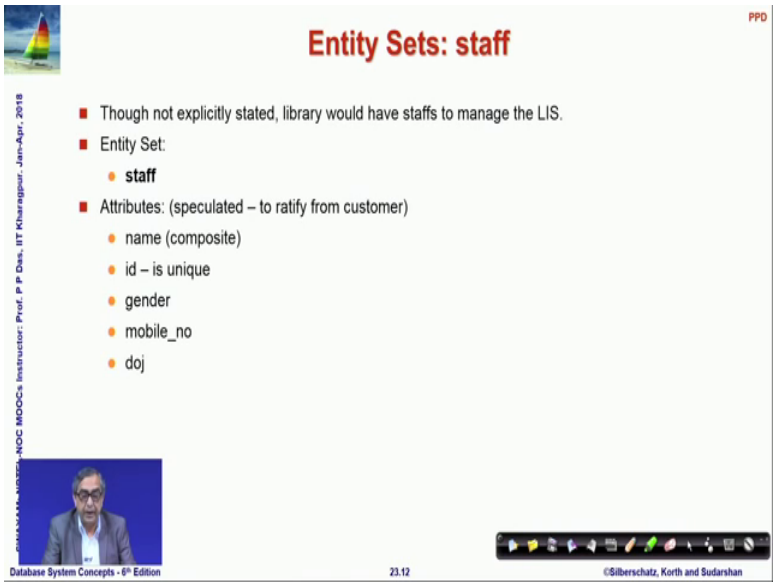
So, let us create an entity set members which has the member number and the member type. So, which can take different types of undergraduate post graduate these different one of these four specific values let us say.

(Refer Slide Time: 10:00)



The rules have talked about having a quota. So, every member will according to it is member category we will have different quota. So, to represent so, quota becomes an entity set. So, we would like to represent that for different member type which is a category; how many number of maximum books can be taken and what could be the maximum duration? So, let us say we will put the maximum duration say in terms of months and with these three attributes we will have an entity set which is quota.

(Refer Slide Time: 10:33)

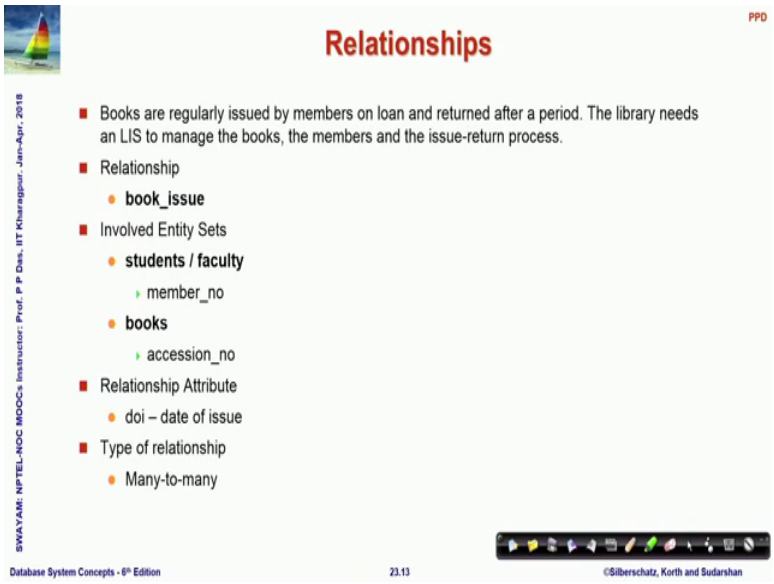


Here, I am talking about another entity sets staff if you again carefully read the specification you will find that there is no mention of this staff in the in the total of the specification, but if we logically think and this is what we often need to do. When we deal with a practical specification like somewhat like, the one that I have given here is that we would need to look little beyond the specification.

So, think about it if I have the library with books students faculty issue process quota and all that then certainly they will have to be some staff of the library; that will actually do all this operation. So, they will be able to log in to the database and actually issue a book return a book check for validity and so, on.

So, let us assume that we have a entity set staff which certainly a staff should have name and id which id should be unique, gender, mobile number and date of joining. So, this is kind of a speculative addition to the design of entity sets and this must be rectified from the customer; when the opportunity arise, but something like that must be there for to make the design complete.

(Refer Slide Time: 11:48)



So, these are ah; obviously, the immediately visible entity sets that we see. So, the other part that we will now have to check on is a relationship. So, again we pick up the relevant statement in this regard books are regularly issued by members on loan and returned after a period the library needs an LIS to manage the books members and the issue return process.

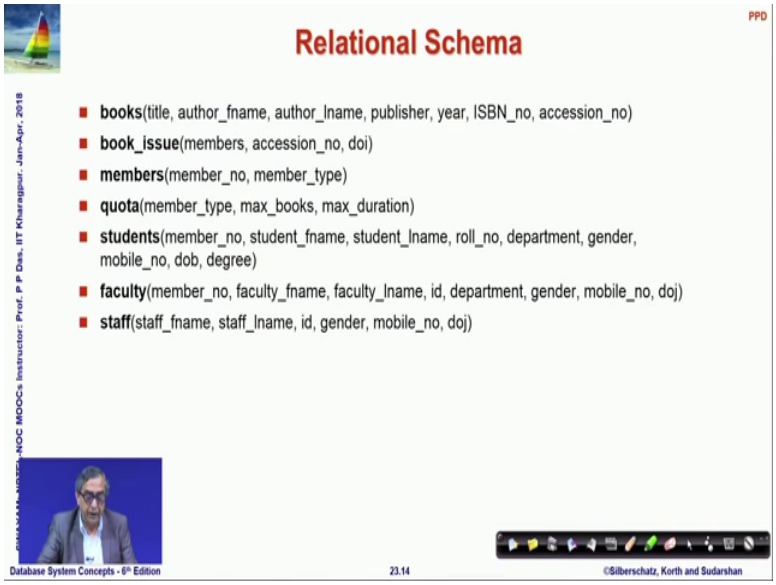
So, certainly there will have to be a relationship of issue between the student or faculty in the books. So, we loosely define this relationship on one side there has to be the books and the other side would be the involved entity set would be either student or faculty so, actually though I am just noting it here as a as a single relationship, but actually there is a relationship of issue between students and books and faculty and books.

So, we will have to see how to handle these kind of situation now certainly the students or faculty are identified by the unique member number of the library that has been given books as we have said are identified by the accession number. Again please note that we are not talking about the ISBN number, because ISBN number could be same for multiple copies of the book, but when you issue you issue a specific copy. So, that accession number which is a unique for a specific copy needs to be tracked. So, these are the two attributes, which will certainly be involved in the relationship and then you would recall that often relationships of their own attributes.

So, here we have one the date of issue needs to be recorded, because we want to check conditions like; if the borrower is has issued the book and how many for how many months he or she is keeping that book. So, if you have to check that we need to know when the book was issued. So, this is a attribute which does not exist in any of the entity sets that are involved in this relationship neither in students or faculty nor in books, but this is a attribute of the relationship which needs to be specified.

And of course, it is this relationship is a many to many relation; because every student or faculty can issue multiple books and every book may be issued by different, but we can we can in general we can be specific to say that this is many to one also if we want to maintain that at a any given instant a book can be issued only by one person. So, if you look at it from that perspective this will not be many to many this will be treated as many to one.

(Refer Slide Time: 14:24)



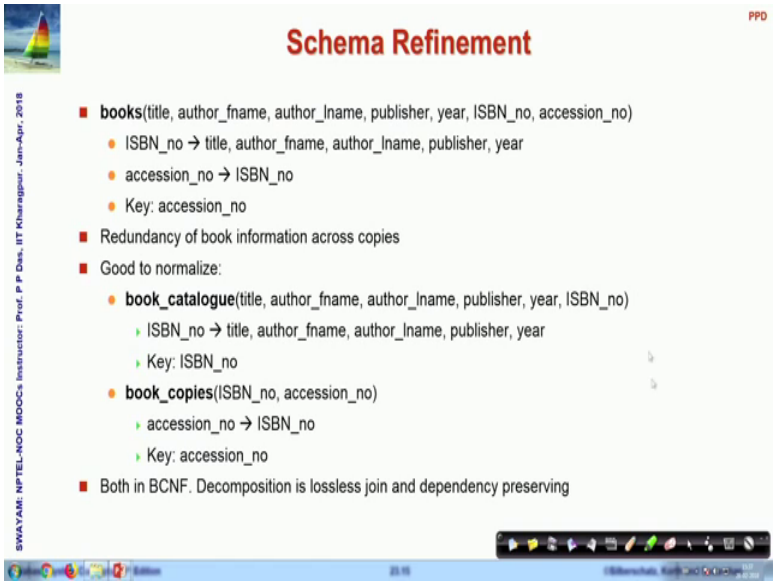
So, having done this finding having found out this entity an attributes and the relationships let us now start with a relational schema.

So, what I have done here; in terms of the relational schema that for each of the entity set. I have created a relational schema by the same name and have put the attributes as identified as attributes of that relational schema. So, this is a very straight forward once we have identification, made this identification process from the original specification. This is a more straight forward process where you just convert every entity set into a

relational schema and also we have converted the relationship there was a there is a relationship called book issue here as you can see the book issue this also we have converted to a relational schema involving the two attributes of member number and the accession number.

So, let us see how this will span out later parts so, having done this.

(Refer Slide Time: 15:38)



The next task would be to work on the refinement of the schema. So, now, it is time that the relational schema is available the first schema is available. So, now, we have to apply all different notions of the relational design to refine this schema and finalize. So, for what will do; we will take every relational schema and we will try to identify the functional dependencies and use that to identify the key.

So, if I look at the books relational schema, then we easily know that ISBN number will uniquely identify the title, author first name; authors last name, publisher and year. So, here also you may you may just note that since name is stated to be a composite attribute I have designed here to use two different attributes the first name and the last name. So, with that we have this functional dependency which tells me that given ISBN number, I know these details, but of course, that does not tell me what is the accession number because there could be multiple copies, but another functional dependency must hold because, every copy of the same book must have the same ISBN number. So, given the accession number, ISBN number should be determinable.

So, accession number functionally determines ISBN number. So, if you do the sample computation on this you will easily figure out that a key of this is accession number. So, this is what we currently have this this is what we have done now. So, you can see that if there are say 5 copies of a book having different accession numbers which are the key they are has been number would may all be same.

So, if I have the same ISBN number I can multiple accession numbers and therefore, there are different records, but if that accession number of two books are I am sorry the ISBN number of two books are same then all of that title, author and first name last name all this attributes will be repeated and if there are multiple copies of the book then each one of them will have a separate entry because they have a separate accession number.

But, their ISBN number and everything else will be redundant. So, we can easily see that the given relational schema actually has redundancy across copies of the book and if we want to look at this from formal theory we can easily check that the keys accession number. So, ISBN number determining title author etcetera that functional dependency violates the voice code normal form actually this is the also not in the third normal form right now.

So, we would do well to reduce this redundancy and it would be good to normalize. So, here I have not gone through the actual steps of normalization, but I am showing you more intuitively as to, how you can normalize; because it is a quite obvious that the accession number is involved only with the ISBN number and which keeps track of the copies.

So, we propose to have under normalization we propose to have one which is let me just highlight and show you. So, book copies where the ISBN number and accession number will be maintained. So, for every accession number we will be able to see the ISBN number which will tell you which book it is and rest of the book details will be kept in this say another new relational schema called book catalogue which will have all of the earlier attributes expect the accession number.

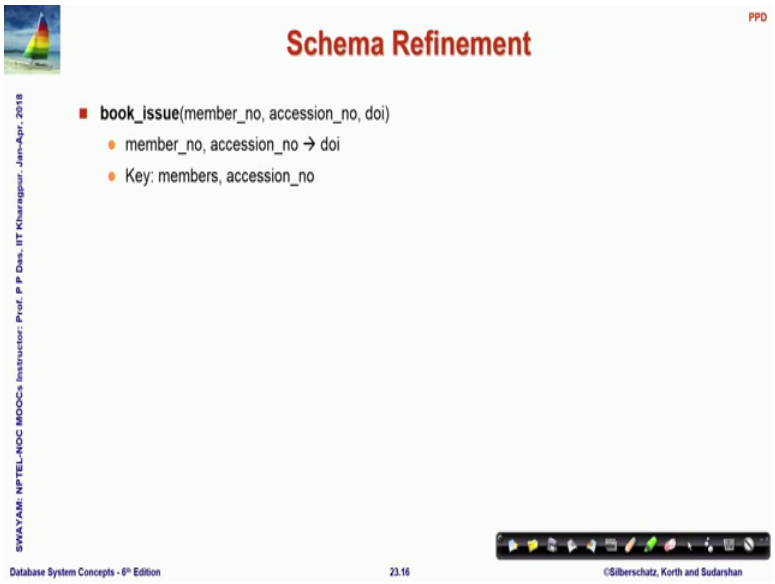
So, now, if you project the; this dependency on book catalogue the dependency will be fully projected. So, the whole dependency is preserved if you project this dependency on the book copies it will also be fully preserved. So, this decomposition is a dependency

preserving and you can easily check that these two can be joint by a natural joint using ISBN number as the common attribute.

So, if we take the intersection of the two sets of attributes then ISBN number would be the attribute and ISBN number functionally determines all attributes in the book catalogues. So, our lossless join condition r 1 intersection r 2 determines r 1 here. So, this will also give me a lossless join and if you check on each one of these relational schema then this functional dependency the left hand side is a key and therefore, book catalogue is in voice code normal form and book copies is also in voice code normal form.

So, you can you could come to the same result by doing the formal process of functional voice code normal form decomposition, but I have just shown you here as to intuitively how you get this. So, intuitively becomes very clear that you have details of the book that you keep separate in a separate table, because that is not change across the copies and we have a separate table to maintain the specific information about the copies. So, that takes care of the books relationship moving on let us look at the other.

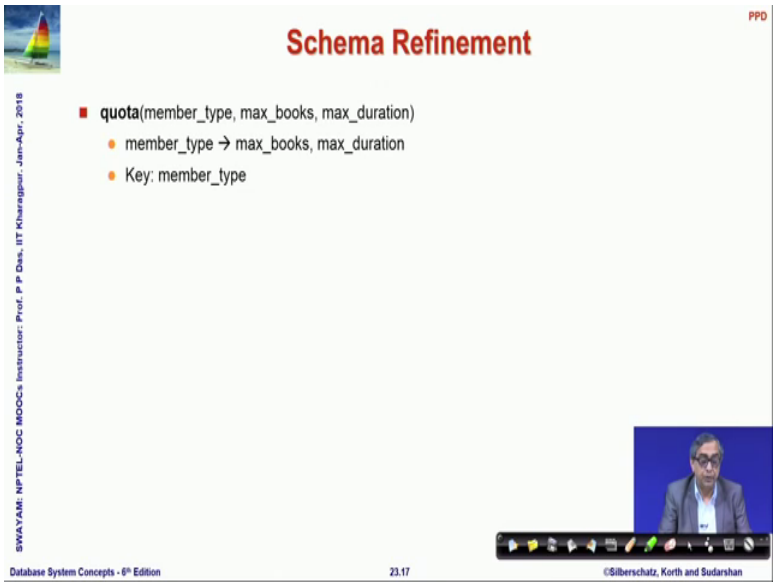
(Refer Slide Time: 21:16)



So, book issue is a relational schema which comes from the relationship between the now between what now book issue is to happen between student faculty and books. So, it is I mean we cannot have two of them of these entity sets occurring in the same. So, initially let us just put that well the library has a concept of a member. So, what if the book issue is a relation simply between the members and the and the books of course, we

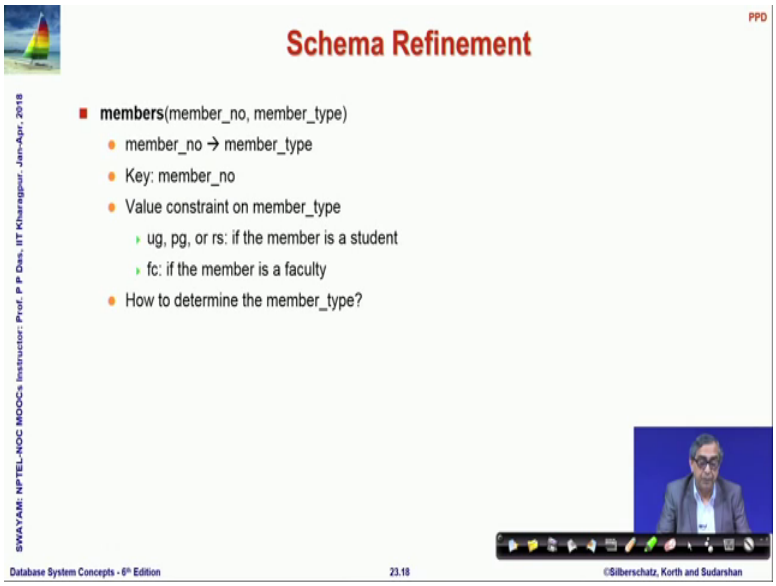
do not know the relationship between members and students or members and faculty, but we can at least refine the book issue to be between member and the book and therefore, member number and accession number together becomes the key which determines the date of issue this is already in normal form as you can see.

(Refer Slide Time: 22:11)



Quota is a simple relational schema where member type determines the max books and duration and that is the key in normal form

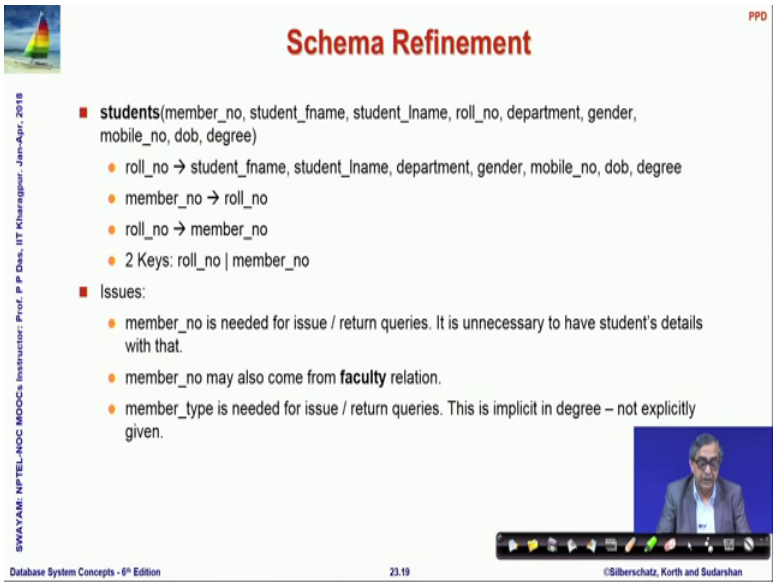
(Refer Slide Time: 22:19)



The members I mean we have identified something like a member which should list all the members of the library.

So, existence of a member number in that list will mean that someone holding that member number is actually a member and it is should also tell me what type of member or what category of member he or she is. So, member number must determine the member type and the member type will be constrained in terms of possible 4, 1 of the four possible values are stated here now of course, we will still have to figure out is to where do we get this member type from and so, on and that information is not present here. So, further refinements will be required in this regard.

(Refer Slide Time: 23:02)



Let us go to the students this is the whole schema that we had done. So, roll number determines all the; attributes specifically roll number also determines the member number and member number determines the roll number, because every student has a unique roll number and; obviously, has a unique member number also number will determine rest of the all attrib. So, member number and roll number both will determine each other and roll utes.

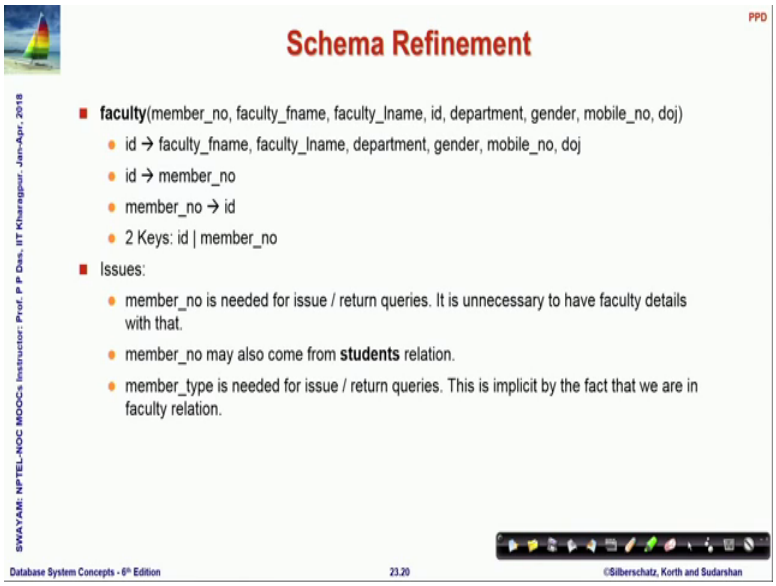
Which mean that this design this relational schema has two keys the roll number and the member number now are we happy with this design that is a question we need to ask. So, we can figure out that member number is needed for issue return or queries book issue has member number, that is; what we were thinking of; now when we want to deal with

this book issue issuing a book to a student and returning a book from the student and so, on.

Is it necessary to have all the students details with that one that make every record very heavy and if we want to extra connect with that do some join or if you want to do some query unnecessarily we will have to deal with very large units of data and as we will see in the next couple of modules that if a record becomes larger dealing with it become naturally becomes more cumbersome. So, there is some kind of discomfort at this design similarly a member number is not here it is the student relation. So, it is coming from the student relation, but in general in terms of issue it may also come from faculty relations.

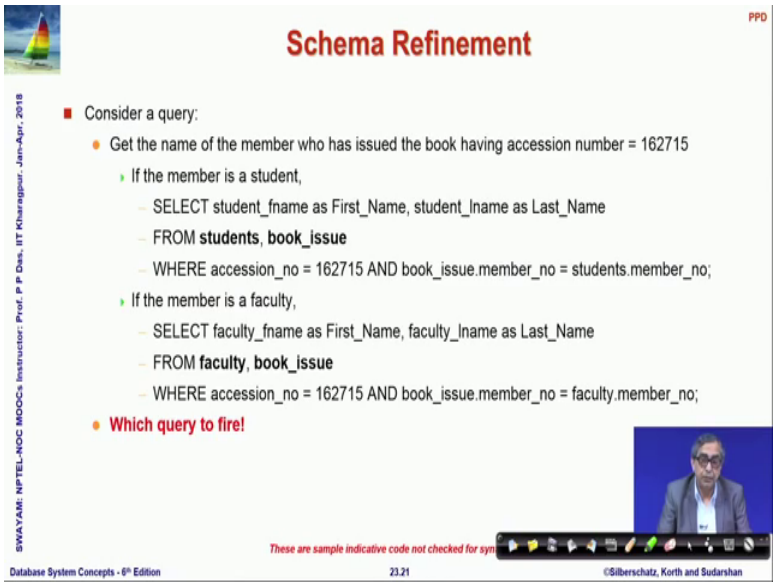
So, how is that handled we do not know then issue return also needs the member type which is implicit here in terms of the field in terms of the attribute degree which tells you that student is a undergraduate postgraduate or research, but it is not explicitly maintained anywhere. So, these are the issues in this form of the design that we that we came.

(Refer Slide Time: 25:06)



Moving on look at the faculty we have a very similar situation as of the student id determines all attributes of the faculty member number is also determine from id member number in turn determines id every faculty has two unique numbers two keys. And we have similar type of issues as we have seen in terms of student. So, we will need to do something in terms of this.

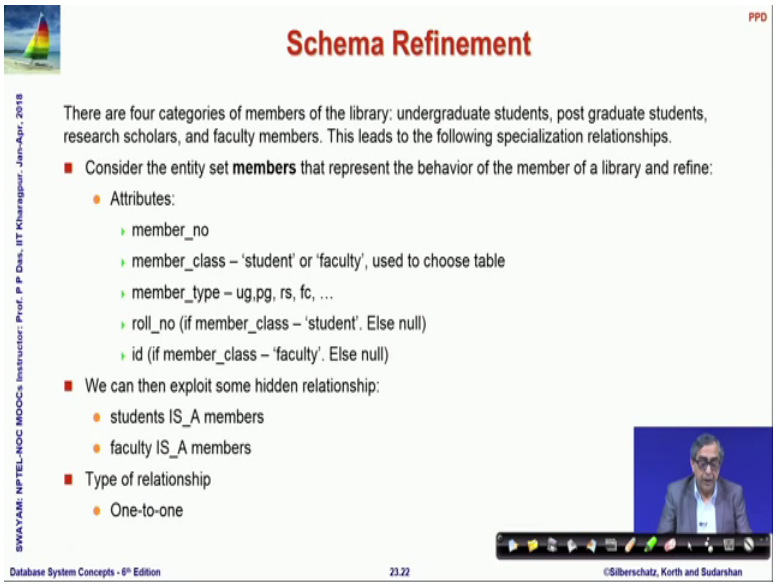
(Refer Slide Time: 25:34)



So, what can us; I mean what kind of issues we will get into. So, let us consider a query that the query is to get the name of a member of the member who has issued a book say having accession number some accession number 162715.

Now, if we have to write a select query for this we will need to know whether the select query should be written on the student or with the faculty. So, if this member is a student then we need the first query where we do a join between student and book issue to find out the student member number who is issued that book where the accession number gives me the particular book issue record from this result will come. Similarly, if the member is a faculty I need a different query. Now, it is a problem because if I have two possible queries and based on the member number I have to decide which query to fire we have not done any mechanism like that. So, this design has problems.

(Refer Slide Time: 26:36)



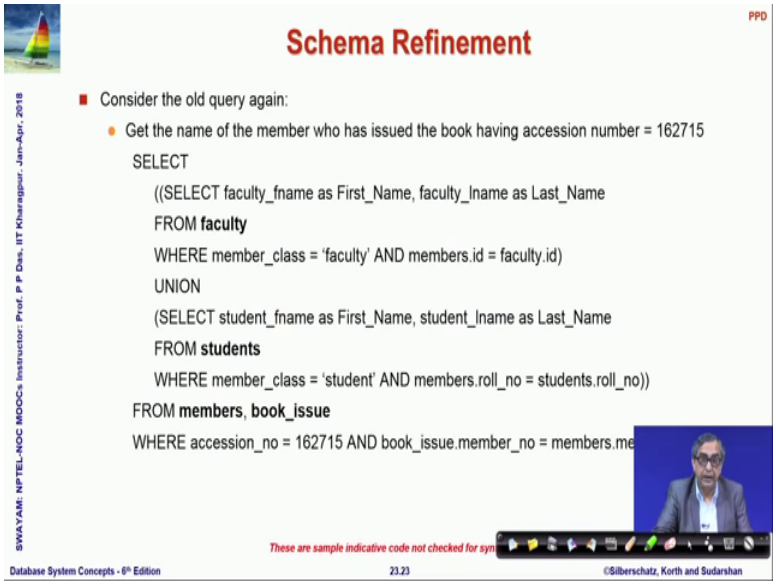
So, let us see what we can d? So, we again go back to the specification and see what the specification says; it said that there are four categories of members undergraduate, postgraduate, research scholar, and faculty members and least to the that least to the specialization of this relationship and we have already done a members concept members entity we did.

Which has a member number and the member type, but now we have just seen that it actually matters as to whether the member is a student or is a faculty is a more unique deciding factor in terms of, how we design our query? So, we introduce a new attribute which was not specified explicitly say; let us call it member class which can take only two values either student or faculty and then retain the member type. So, what it will mean that the; if the student class is student then the member type could be u g, p g or r s and if the member class is faculty.

Then the person is a faculty than the member type should be only f c and then also maintain the roll number and id in this members table. So, the roll number ah; obviously, if it is if the member class is student then the person is a student and the roll number will exist, but the id which is an employee id will be null and at the same time if member class is a faculty for a record then the roll number will be null because, a faculty cannot have a roll number, but the id will be present.

So, we extend the members entity set in relational schema with these attributes and once we do that then we kind of exploit a hidden relationship which was not very explicitly stated anywhere that; now we can say that students is a members faculty is a members, that is; from the perspective of issuing books and using the library both students and faculty can be consider to be members it is kind of a you know virtual entity that we can see here and certainly there will be a one to one relationship between the students and members and faculty and members.

(Refer Slide Time: 28:39)



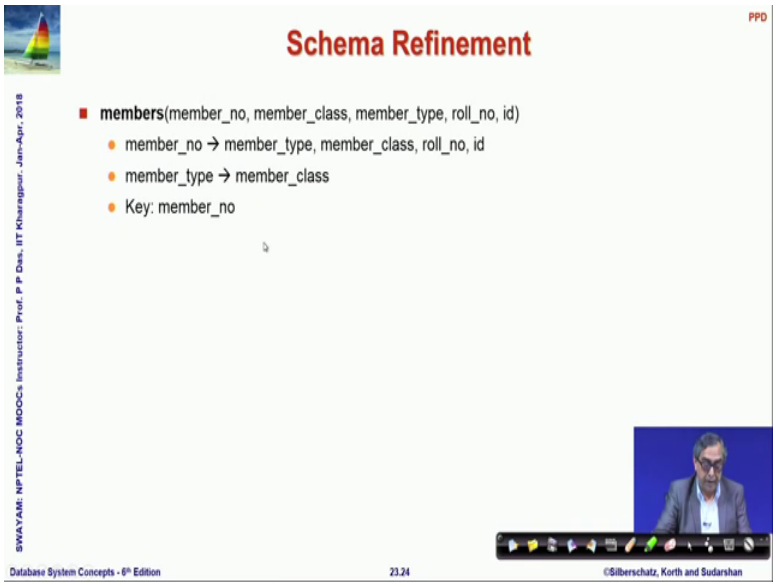
So, let us see what is a consequence of that; in terms of the earlier query. So, we look at go back and look at the query again we will see that problems have got significantly solved, because we now still have to find that member name and this is the basic condition which say.

But, I am sorry this is a basic condition which say the specify the member number who has issued that book, but the actual problem of finding the name can be solved here, because I can I write two queries and take a union of the first query runs on faculty the other query runs on student. Now, here since I have a member class which tell me whether the member is a faculty or the member is a student. So, one of these queries will actually return a null result will not written any record because it will not match this condition, but the other one we will match and will give me the record give me the corresponding names first name and last name of the member and certainly to ensure that

when we are looking at the faculty. We need to check the equality of member id between the members relation and the faculty relation and while dealing with the student we need to check for the equality of the roll number.

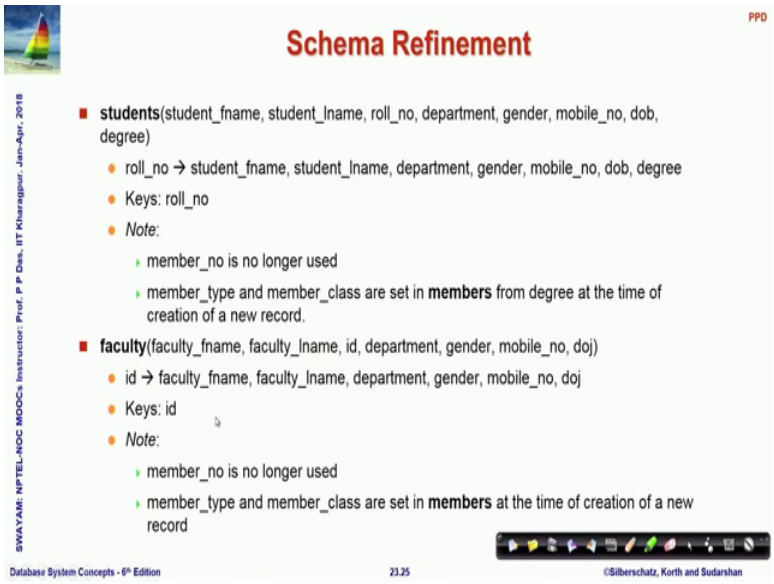
So, in this way by using intelligently using the union feature and the you know nested query feature we can easily write a query and where implicitly. Now, based on the data the switching of which table I am I am actually looking the data from will get solved.

(Refer Slide Time: 30:17)



So, with this refinement my members schema now turns out to be member number member class member type roll number and id member number determines everything it member type also determines member class, because if I know some member type is undergraduate I know member class is student and so, on key; obviously, is one number.

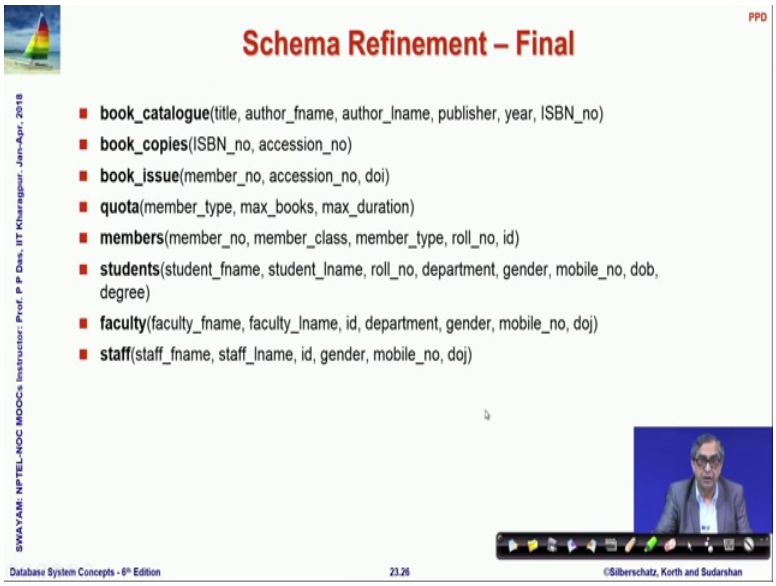
(Refer Slide Time: 30:38)



So having done that, we again go back to the student and faculty, because now we do not need member number in that anymore; because these will not directly be involved in the issue return, because all that you need is just the member number which is already there in the members. So, now, it is a roll number which determines everything member number is no longer used.

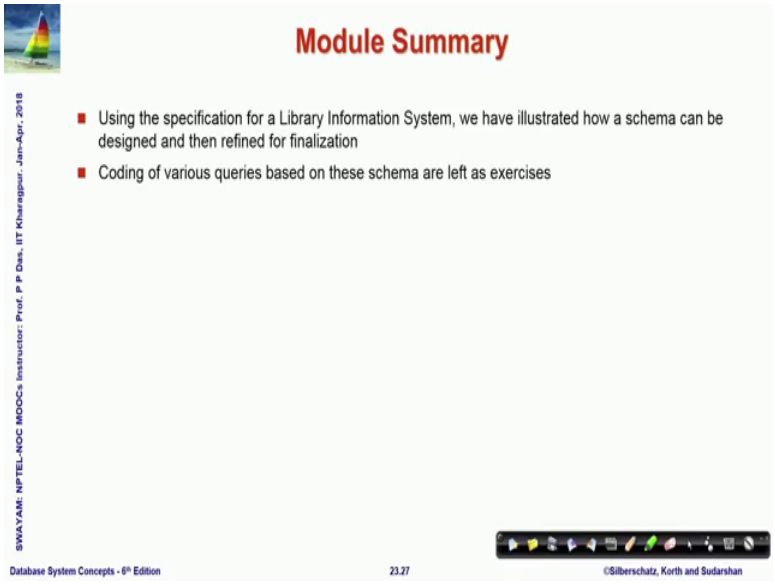
And member type and member class which we have assumed is exist in the in the in the members will have to be derived from the degree value at the time when a new record is created. So, when a new student record is created an entry will happen in this relation as well as a corresponding entry we will need happen in the members relation where the membership number is given and the membership type will be derived from the degree similar change will happen in terms of the faculty as well.

(Refer Slide Time: 31:31)



So, after refinement now we have these 8 relational schema from book catalogue to give the details of every book copies which keeps the information about, how many; what are the different copies book issue; is the basic issuing information. We have quota members has the virtual kind of relation that we have created to support the notion of members of the library and students and faculty are related to this members either through roll number or through id in a selective manner and staff we have not touched.

(Refer Slide Time: 32:12)



So, this is the final relational schema. In this module I have shown you illustrated you that how starting from a very simple specification you can reason and work through in terms of creating the entity sets attributes and relationships and then get into the relational schema look at the possible functionality dependency and refine that and also look into what will be the requirements of doing your query and come to a final refined schema.

So, various queries that we had talked of and others can be can now be coded on this, but I leave that as an exercise to you please try out coding those queries and you will be able to learn in terms of how to do that well and I will try to also supply some supplementary information on this offline.