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

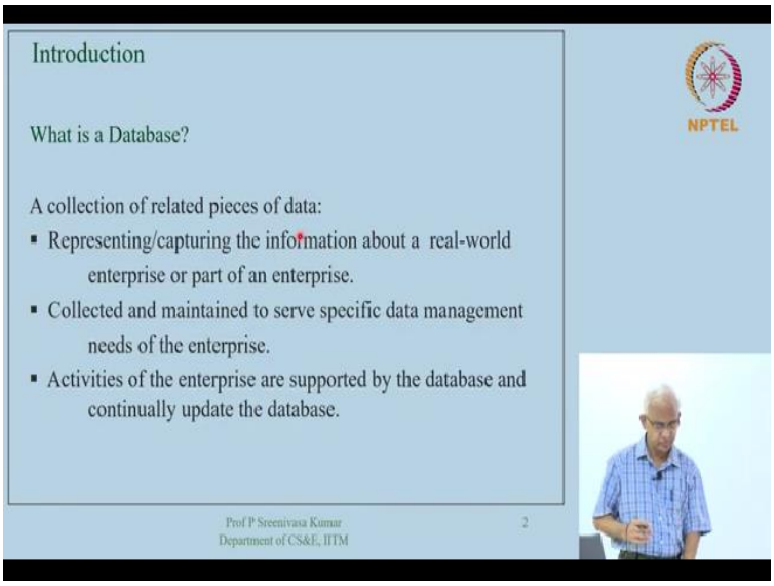
Lecture-01
Introduction

Welcome to the 2019 edition of the database systems course. My name is Sreenivasa Kumar. So, this course is split into several modules like about 8 modules are there. There is an introduction and then the main module is about the relational model. And we also have what is called the entity relationship model module and then we have tuple relational calculus module and SQL.

The database language standard and then we have a module on file systems and physical representation of databases. And we will be spending some time on designing databases by studying what are called normal forms for database designs. So, that will be a separate model. And then finally towards the end, we will discuss the techniques of query optimization. And then the last module will be transaction processing and error recovery.

So, this is how the course is split into several modules. So, in a module there will be the several lectures and we you know we may not be able to, we will try to keep the lectures such that we do not start off a new module in the middle of a lecture. Okay, so let us go ahead with this course, it is a very interesting and fundamental course for computer science curriculum.

(Refer Slide Time: 02:29)

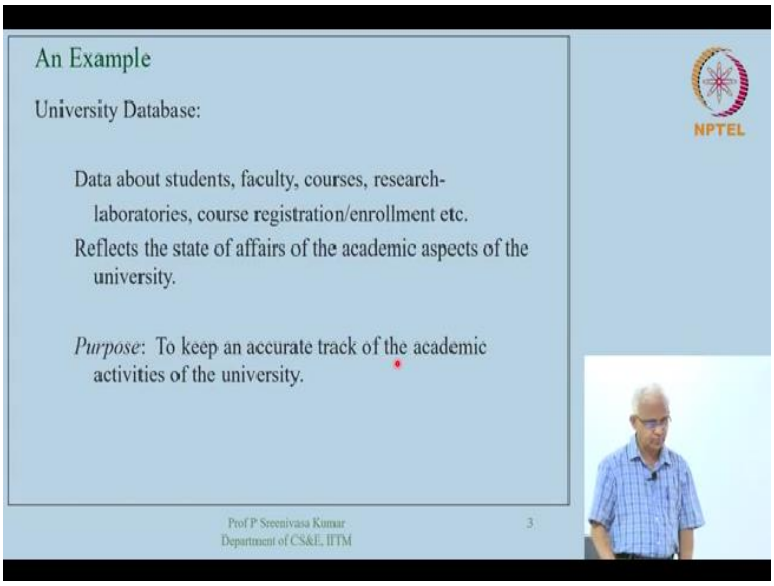


So the course is largely about relational databases. So let us begin the by asking, what exactly is a database. Now, database is a collection of related pieces of data, that represent or capture the information about some real world enterprise are part of the enterprise. So sometimes an enterprise might be so big that it might actually require multiple databases to capture its complete information.

So it is related pieces of data that I think is easy to understand, because supposing we are talking about, as I said a university database, suddenly we do not find weather data in that, right. So it is it has to be all related. Now, it is the data is collected and maintained to serve some specific data management needs of the particular enterprise. So there is a purpose behind building that database.

And the day to day activities of the enterprise are actually supported by the database. And they will be continuously updating the database. Database is a collection of information. So as the enterprise works and changes happen in the enterprise, all those changes will get reflected in the database and they continually update the database.

(Refer Slide Time: 04:28)

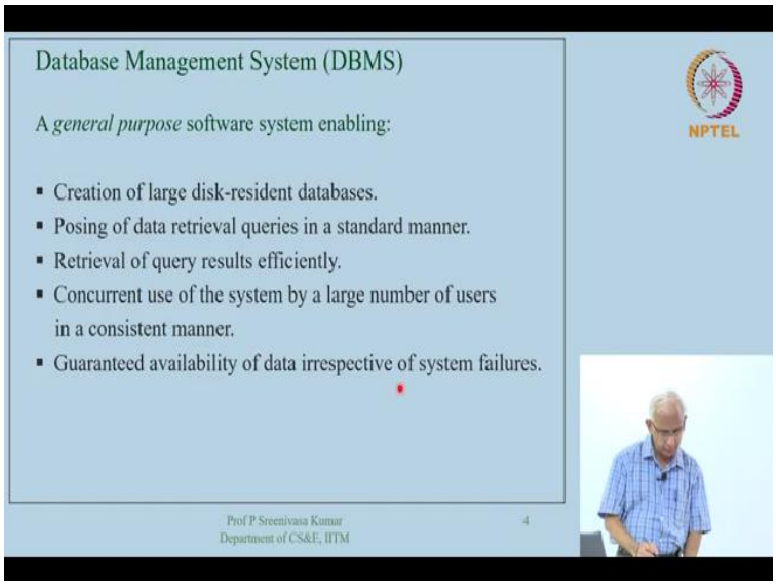


Now, let us take an example. Say, university database. So the information that we would like to collect in this context is of pace data about students, faculty, who are teaching the students, the various courses that are available for study. The course is a subject and then research laboratories, their information about them and the process and the enrollment or registration of students in various courses, all this information would like to keep track of.

So let us say our database reflects the state of the affairs of the academic aspects of the university. That means we will, there are some, let us say, the there are several things that we need to keep track of a university, but this particular database takes care of only the academic aspects of the students. So the purpose is as I was just mentioning, to keep an accurate track of the academic activities of the universities.

So this is an example situation where we will require to build a database and then operate the database and then keep track of the various activities related to the academics of the university.

(Refer Slide Time: 06:10)



Now let us first distinguish between 2 things. A database and a database management system. Okay, a database is as I was just now mentioned, collection of related pieces of data. Now this collection of related pieces of data could actually be maintained even in physical ledger's, books, right. One could maintain that. But then we were talking, we are talking about how do we reverse this whole thing, and then maintaining this on computer systems, and that is where a database management system will come into picture okay.

A database management system is a complex and in general purpose software system. It is a software pretty complex software is that, which will help us create large disk resident databases, large disk resident, disk resident is important here because this are the medium which gives us the ability to store data for a long period. And of course, disks do fail. So we will have to keep track of we will have to handle that issue and separately, we will also discuss that.

But secondary storage is the mainstay of databases, because the primary storage of memory is you know, volatile. And as applications you know, close the information stored in volatile memory will vanish and whereas for databases, the information has to be maintained for a long period. And we need disk resident databases. So DBMS helps us in creating these large disk distant h databases.

And it also helps us in posing data retrieval queries in a standard manner okay, so once we have a database would like to retry information from the database and make use of that information for various purposes right. So has the student in case the third semester, done this particular course and apply integrate or not, would like to figure out that. So how do we get the information is there in the database.

And we need to pose a query to the database and get back the answer from database. Now DBMS and in general, the relational database theory, you know enables the process of posing this particular query in a particular standardized manner. And you are all probably aware of that standard, which is this SQL standard. SQL is not only just language, it is in fact the internationally recognized standard of how to pose queries.

Okay. So, the database management system supports the, the standard. And it is the systems work to kind of ensure that the query results are retired efficiently. That means in as short time as possible we would like to get the results. So this will depend on how much data is actually being stored in the database. Okay if there is a huge amount of data is actually stored in the database, then retrieving a particular piece of data might in fact take a lot of time we do not.

So, it is in this context that efficiency of retrieval of data comes into the picture. So, the database management system strives to you know always strives to provide efficiency while running queries efficient solution. Then, another important aspect is the concurrent use of the system by a large number of people in a consistent manner okay. Now, a database system is meant to be not just to be used by one person.

It is in fact, used to be used by a huge number of people and it turns out that it is much more efficient for us to you know run these requests from the end users about for information from the database in a concurrent manner, that means the end users will get an impression that the database is actually answering only his or her question but it is actually currently handling several hundreds of people.

It is inefficient to you know make the database server answer one question at a time. Because, you know, we will later on see why it is so, and so, we would like to make the system you know, answer queries answer end users questions in a concurrent manner and while that is happening, maintaining consistency of the data is an important aspect. Now, what exactly is a consistency requirement will depend on the actual application or actual database right.

So, for example to give an instance of this kind of thing in a railway reservation system, it major consistency requirement is that at no point of time a seat is located to 2 people, a seat between one point A to point B on a particular train should be allotted to exactly one person, it should not be allotted to 2 persons right. Now, imagine this situation in the context of these seat requests coming from concurrent users hundreds of users at a time.

And maybe several of them are asking for seats between point A and point B and in that context, we should ensure that the system never allocates a particular seat to 2 different individuals. So, like that consistency requirements will change from database to database domain to other database to database and those things have to be kept in mind while we are concurrently running the large amount of these requests large number of users.

Now, another important aspect of this databases is that it is the guaranteed availability of the data irrespective of system failures. Now so what kind of system failures are we talking about, system failures as several kinds of system failures are possible. One of these let us look at them, let us let us pause a little while and then see what are the system failures that we have to handle. First thing is the storage medium can fail, the disks on which way of storing information may fail.

Your wall probably, you know, keep backup of your data on the laptop, because the laptop disk can fail right. So, disks do have this failure probabilities and discs can fail. So, but a database management system which is handling the data of a large enterprise, you know, has to be careful and it cannot say that my disk has failed and so I lost your data, right that should never be allowed.

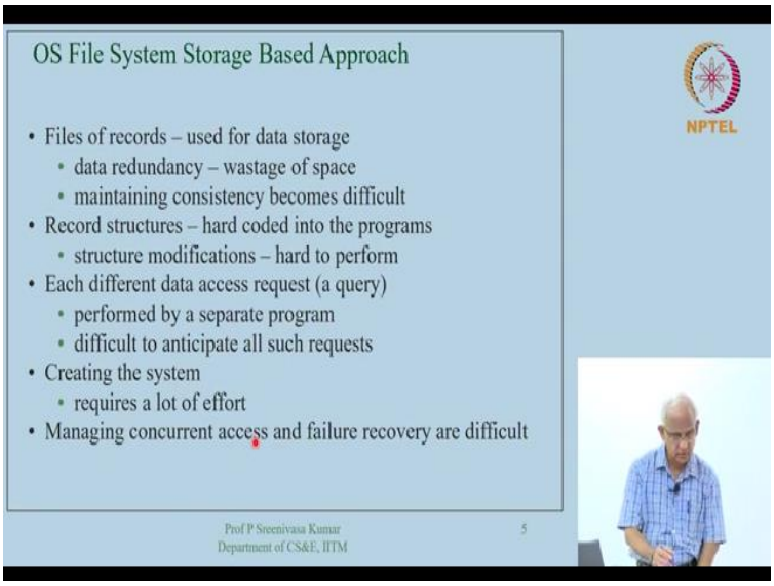
So data should be permanent guaranteed that it is going to be available irrespective of system failures. So, we should take appropriate measures in the DBMS database management system such that even if disk fails occur the data will continue to be available. Now, enterprises based on the kind of database management system we are talking about and the various components in it we have to know there are several measures that are taken in order to handle this issue.

Like for example, you sometimes replicate the entire data in a database in a geographically different location. So, that even if there is a catastrophic failure like a fire or you know, storm or something like that, and a database physical database server is damaged beyond repair. The data is still available in a geographically different location. So, we have to take this kind of measures in order to ensure that the data is

Now those failures are what are called catastrophic kind of failures that were in occurrence. But then more often what happens is that these processes that are running on the database, you know, might encounter some errors and might crash, a programs might crash. And so and if the program crashes, if a particular application, if a particular process that is doing certain changes for some reason, crashes.

Then if it has changed this database and then could not complete its work and crashed, then we should also ensure that exchanges are not really reflected in the database because it might have done some partial work on the database and such thing might actually affect the consistency of the data. So, irrespective of system for use of several kinds, we will in fact, discuss this topic in much detail when we go to transaction processing and error recovery module of the course.

(Refer Slide Time: 17:24)



Now, one question that you might ask is that why do we need a special purpose, you know, software, this DBMS software, why cannot we actually we are all you know see the language C or C++ as the case may be which are pretty powerful and you know so, they are capable of handling files and then you know after all information is all nothing but structures or records and then we can create files of records, sorry, files of records.

And then store this information on individual files. And we can then write programs to manage these files of records and then provide this kind of capabilities all by ourselves in a programmer who is capable of handling you know, files on the operating system provides file system. And so we should be able to write programs that will you have probably tried that out in, in a data structures kind of course, where you open files.

And then keep track of the structures and such through records and then retry records and all that. So why do we go for this special purpose software. Why cannot we do it ourselves, now first thing is if you go for files of records, maintaining consistency becomes difficult. Okay, so let me explain this a little bit. Let us imagine the 2 kinds of you know, database data management requirements.

Let us say one application wants to keep track of the students, their courses and grades. Another application likes to keep track of the students you know information about the health records of

students, let us say the institute hospital requires that kind of an application. Now, if certain basic information about student is replicated in both places like the students name, contact number, addresses and such details are obviously you know to be replicated in both places.

So, that leads to a lot of redundancy of data, data will be stored in 2 different places in a redundant manner. But nowadays, redundancy of data is not a big issue, I mean we have large disk space discs and you know we can maintain them. Not a big deal actually to how a lot of information store your smartphones have 64 or 128 GB of space nowadays, right. So, this space alone is not an issue.

But what is an issue is, if you store a piece of data in 2 different places, if a change occurs, then in order to keep this information consistent, you have to necessarily change it in all the places where it is store right. Otherwise, there is inconsistency in data right. So, according to that particular set of records, his phone number is this according to this another phone number is there and which is actual phone number you do not know right. So, it all has to be updated in all the places and updating and maintaining this if you write your own independent programs becomes really difficult.

Because this requires communication between the operators of those specific programs and all kinds of problems are possible here. Now, another thing is that these record structures that we create in order to keep track of information will have to be hard coded into the programs, right you have in when you write essays program or a Java program, you will first declare where he was right.

So, the structure before you can create a file of the structures, the structure has to be defined you have to give a declaration of the structure and then create variables of that particular type and then create a file of that particular kind of records etc. right. So, these wants to hard code them into this programs any change any slight modifications in these structures will lead to recompiling your software.

You have to recome, you have to change the definitions and then recompile and recreate a new executable and then run the program again. So, the structure modifications become hard to perform. Whereas we will see later on how DBMS approaches this issue. Now, most important thing is that we in the last slide also we discussed about queries. Now, if you write your own programs, then how do you handle queries.

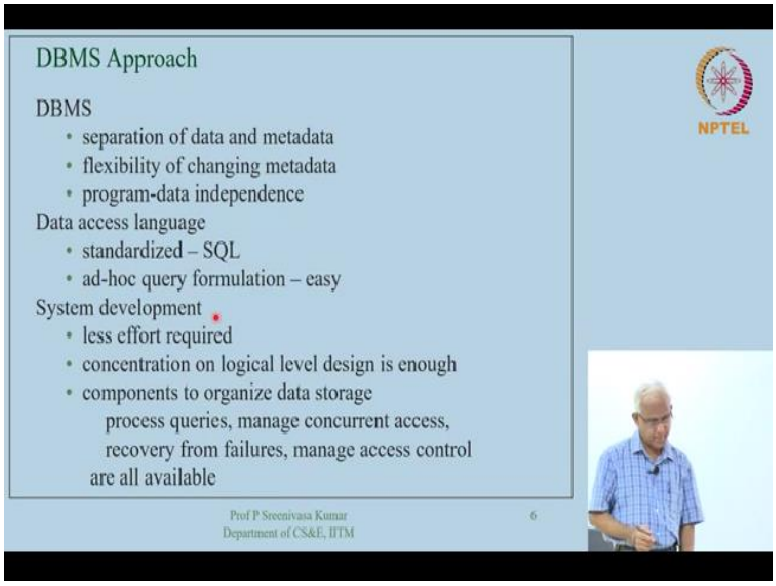
How do you handle queries, how do you anticipate what are all the kinds of queries that come each because for each of these queries, you have to write a special purpose program because you have the set of records with you. And so if you have to process a query, then you have to write a special program, which will take the requirement of that query and then search through the records and then give out to the answer.

That is the only way you can handle queries. But then how, how many different queries or can you write programs for the different kinds of queries. So it is very difficult to kind of anticipate all such queries and then write programs for all of them and then keep them ready. So that the when the end user says that took, I want this information you can read, okay, go and run the you know program 10 it will give you answer.

So it is impossible to do that. So, creating any database like system, you know by writing our own set of programs is a very requires a huge amount of effort, and it is almost kind of impossible to achieve the requirements of a database without going for a database management system general purpose software okay. So, the DBMS approach of course another thing is this managing the concurrent accesses.

And then recovering from failures. This is also becomes a programmers responsibility which is again concurrent accesses, managing concurrent accesses is a very difficult task because you are almost like you know doing a job of an operating system in that thing okay.

(Refer Slide Time: 25:39)



So, because of all these reasons, we will actually not attempt to write our own set of programs in order to handle the requirements of a database. So, the DBMS approach this entire issue of creating and managing databases is very interesting. First thing is it does the separation of sorry separation of data and metadata. So, the metadata which is this structure information, that means all those you know various components that go into structure definitions. These are maintained separately in what is called a catalog of the system.

In a catalog, we will separately maintain the record types and separate them from the actual programs that handle this records okay. So, this gives us flexibility for changing metadata. So **so** this aspect is also what is called program data independence. So now how does this help the members, the software that, the programs that you will write allow become independent of this structure, what it does first is to okay.

If you have to handle as a certain kind of information, it will first go and check in the catalog as to how the information is actually stored in a file okay get that information and then start operating that particular file according to that metadata, that has been supplied about that particular and this all happens internally. So, programs become in some sense independent of the data that that they are handling okay.

So, if there is any structural modification that has to be done, it can be done at the catalogue level, by a separate program and then those programs that are handling these information can again read this modified catalog information and then go about handling the actual data records okay. So, that way we get what is called program data independence. Because of this, the DBMS becomes general purpose.

Because a DBMS in typically can be used for creating different databases, one database for say hospital records, one database for staff accounts and salaries in Institute, one record for one database for the academic records of students etc. okay. So, but one DBMS system will be able to handle all these 3 or 4 different databases. Okay, so for each of these databases there is a separate catalog information.

And so the DBMS first opens the catalog of how the information about the database is actually stored. And then most about manipulating the actual information in the database that is how it achieves general purpose ability. Now, SQL is a standard. So, SQL has resulted after this relational database theory has been large has been adopted for constructing databases.

And SQL is a worldwide standard and as long as you know implement all the SQL semantics then we can actually handle any kind of query. So, query formulation becomes easy because this language standard SQL will help us express our data requirement queries and all the DBMS has to do is to basically implement the structures of implement the constructs of the SQL okay.

And so, it kind of you know, handles all kinds of queries and SQL is internally you know is based on what is called tuple relational calculus and also we will be studying what is called relational algebra. All of these things are there inside the box, so to say and once you establish that these languages are capable of representing all kinds of queries. All that the DBMS has to do is to implement you know the language.

Once the language is implemented or the standard is implemented, then all this ad-hoc query formulation becomes easy. So, that is why that is the major advantage of going for a DBMS software. Now overall system development of a particular database. Take for example, let us

take this the academic aspects of students. So it requires us to sit down with the academic section of the institute or the university.

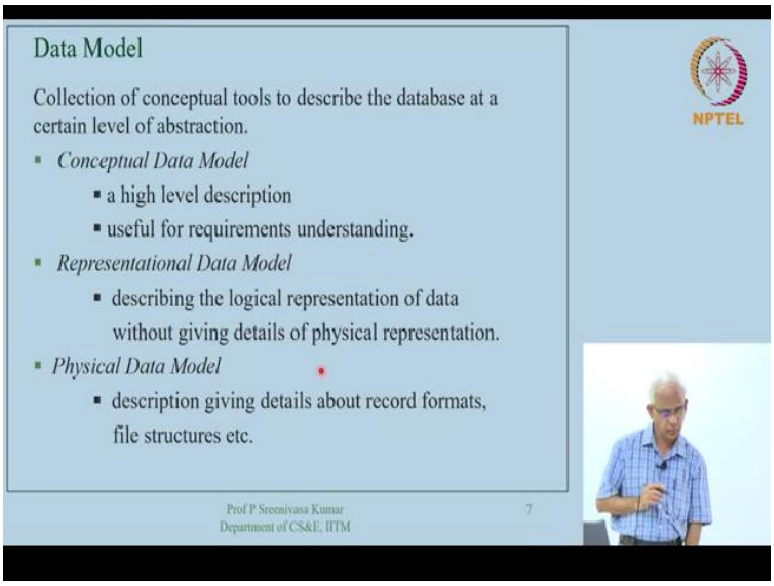
And then find out their requirements as to what are the various things that they would like to keep track of, and then, you know, come up with a logical level design and we can actually, because of this availability of DBMS we can actually concentrate on the logical level of the design and then we can kind of implement it on top of this with the help of this DBMS.

So, all these components to kind of organize the data storage, the processing of queries, the management of concurrent users, all this is automatically taken care by the DBMS. So, all that the designers have to focus is now on the logical design of the database itself, how what kind of information is represented and what kind of standard queries are required to be run on a day to day basis, etc.

All these things if you decide then we are all set to kind of set up a database. So actually some of these things are you know, unless you really think about it, you may not realize it because you might be actually using some web based you know, in tools that will set up the databases for you and things like that, right. So unless we really think a little while, we you may not realize is how life was without a database management system.

Because now, people take it for granted that if you have to set up a database, you use a database management system and popular database management systems are Oracle is you know DB 2 IBM sleeby 2 and Microsoft's SQL server and so many other database management systems are available okay. Any questions in order we have been talking about okay.

(Refer Slide Time: 34:04)



Now, let me bring in a new, one more interesting one interesting concept, the concept of a data model. A data model is a collection of conceptual tools, conceptual tools to describe the database at a certain level of abstraction, computer science is a science of abstractions, we create abstractions and make use of them, right. So, I will actually tell you as to why there is a need for describing the data base at various levels of abstraction.

But first thing is to understand that there is a collection of these conceptual tools that will help us in describing the database in a at a certain level of abstraction, we will talk about data models in detail now. Now, various kinds of data models are possible. One of them is conceptual level data model. A conceptual level data model is in some sense a very high level description.

It enables a very high level description of the database and it is very, very useful when we are trying to understand the requirements of the database and trying to understand the requirements of the database. So when we go and as a team of database designers, when you go to the academic section and then talk to them as to find out how, what kind of information you would like to keep track of etc.

It is this model that we will try to use to gather the requirements, we will go into, in fact, there is a whole module on one of the popular conceptual data models, which is this entity relationships model. We will talk about it in the next module. Then there is this representational data model .

This describes the database at a logical representation level without really giving much details about the physical representation.

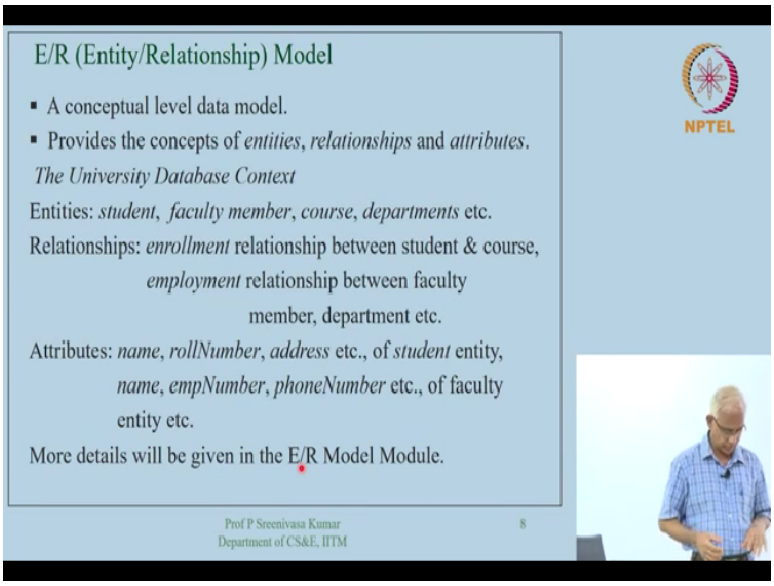
Ultimately, all data is physically represented you files of records okay. Physical representation at the lowest level is in the form of records files okay and certain external data structures, you have studied data structures which are in memory data structures, in this course will look at discrete in data structures so okay. So with the collection of files, records, discrete in data structures we will actually physically represent the database.

But at the in a representational data model we will not really bother I will hide all these details, will hide these details and then only talk about a logical representation, how logically how data is represented in a logical manner. That is what we will describe, this helps us in focusing on the design, if the design is not proper, then we might actually get into a lot of problems. And so, we would like to focus on the design of the database.

And we can do that with the help of this representational data model. Then the physical data model is the one that actually describes all the full details of the various record formats and then the file structures, what kinds of files are there and what kinds of external data structures are there etc. and how are they related to each other all that information is to be represented and that will happen at the physical data model level okay.

So now we will take one by one I will give you little bit more detail about the conceptual data model and the representational data model in the next few slides, any questions.

(Refer Slide Time: 39:16)



Okay, one of the choices at the conceptual level data models is what is called entity relationship model. What it does is to provide concepts of what are called entities, relationships and attributes. So, these are a little bit general kind of concepts, which we can explain to non computer science people, non technical people, we can quickly explain these concepts to them.

And then sit with them, discuss the database you know database description, what kind of data you want, how what are the various entities to be kept track of, what are various things to be all that we can discuss in terms of these concepts, we will engage in a discussion and then gather requirements for the database. For example, if you take the same university database context.

The entities the major things that the database has to deal with, could be a student, faculty member, course, departments, etc., these are the various things that it has to handle and then the relationships that means, associations between these entities, for example, is the enrollment relationship between student and course, students enrolled into courses, the employment relationship between faculty and the department.

Each faculty is employed in a particular department etc. So, these are the kinds of example associations or relationships and then various attributes, there are various attributes like the names, role numbers, specific you know role numbers given to students to identify one student

from the other, the addresses, the address of the student etc., this could be the some of the attributes of the student entity that we would like to capture.

So, we will give more details about this entity relationship model in a separate module that we will take up just after the introduction okay. So, at this stage we takeaway point is that there are this conceptual bag of tools using which we will describe the database at a very high level and sometimes at a logical level and at sometimes at the most detailed level. So, we need tools to talk about these things with the descriptions of the database itself.

And those things are what are called data models, okay. So, I will in the next class, I will talk about the remaining kind of data models, will stop today with this okay, thank you.