Geographic Information Systems
Prof Bharth H Aithal
Ranbir and Chitra Gupta School of Infrastructure Design and Management
Indian Institute of Technology - Kharagpur

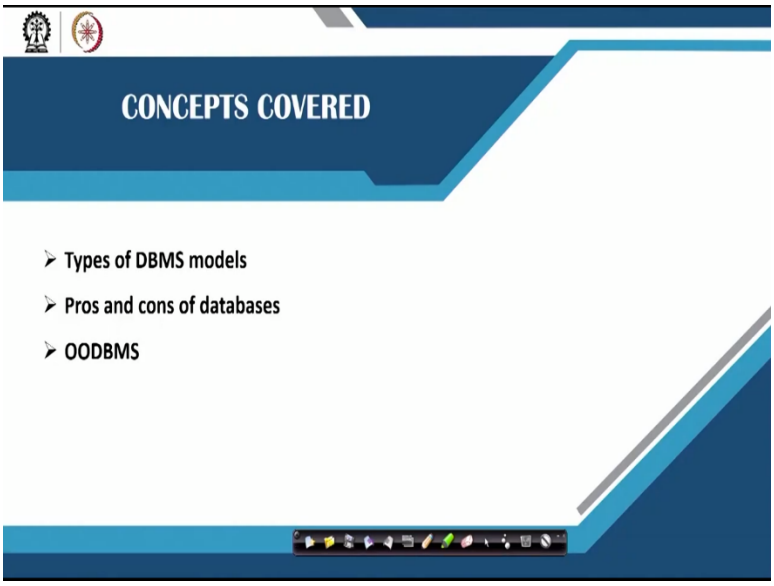
Module No # 07
Lecture No # 34
DBMS Models

Hello Namaste welcome back to the course on geographic information systems. We were discussing about database management system. We looked at different database we looked at data definition language, data manipulation language. Then in the previous session also we looked at what is normalization and what is a need for normalization and a goal for normalization.

And we looked at different features or let say what do you mean by key the primary key etc., we looked at all of those aspect. In today's class we would look at different models of database management system. So there are different thinking different aspects of how the database management system is handled or database is handled or managed. So we would look at all of these aspect in a or I will give the representation portion of the models which are exist in terms of database management system in today's world.

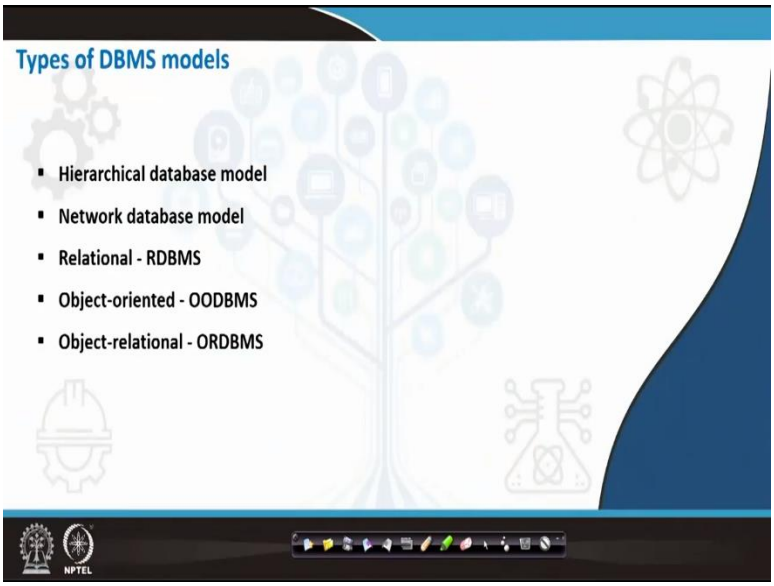
But these are essentially not covering the entire aspects of what do you mean by database management models and DBMS models but just a represented a factor and just a information that you may need in case you are trying to look at more advance things in GIS ok.

(Refer Slide Time 01:42)



So we would look at different database management system model look at pros and cons of different databases then we would also get into what do you mean by object-oriented database management system. So we would also look at that as a last concept in this particular lecture.

(Refer Slide Time 02:02)

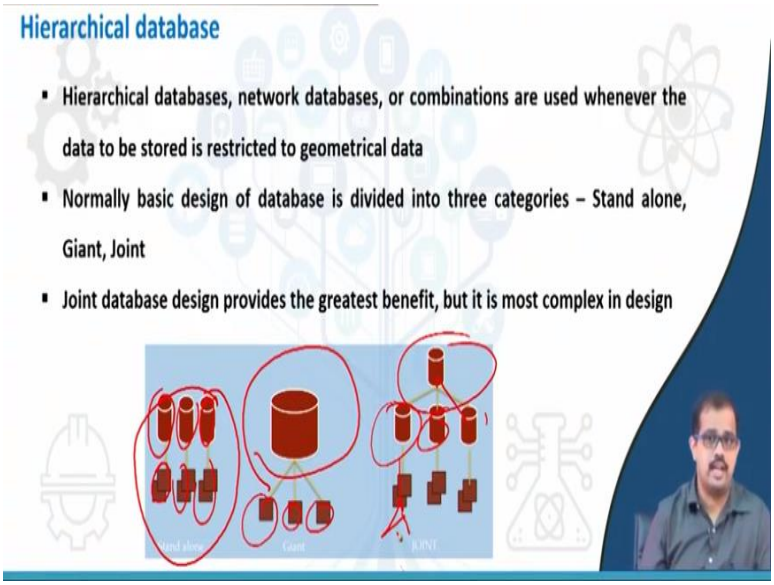


And when we look at different database management models that are available today there are hierarchical database model, there is a network database model, there is a relational database modeling system or model system, then you have object oriented database management system or a modeling system, then you have a object oriented relational database management system.

So these are very well know type of different database management systems. So there are huge number of developed more interesting way database management system which probably you can even look at on the web but as of now when we are looking at geographic information systems. So these are the very well-used database management system but RDBMS is one management system which is well recognized and used by huge amount of users for their applications.

So relational database management system is one that we would look forward when we are looking at the entire GIS package. So normally that is a one that is mostly used where in many of the GIS package is that operated in today's scenario ok.

(Refer Slide Time 03:18)



So the first type of database is the hierarchical database that or the hierarchical database management system. When we look at hierarchical database network databases or combination of these are used whenever the data is to be stored is restricted to a geometric data. When I say geometric these, I am very specific to the geometry of a data ok. Normally basic design of a database is divided into 3 categories one is a standalone, giant and another is the joint.

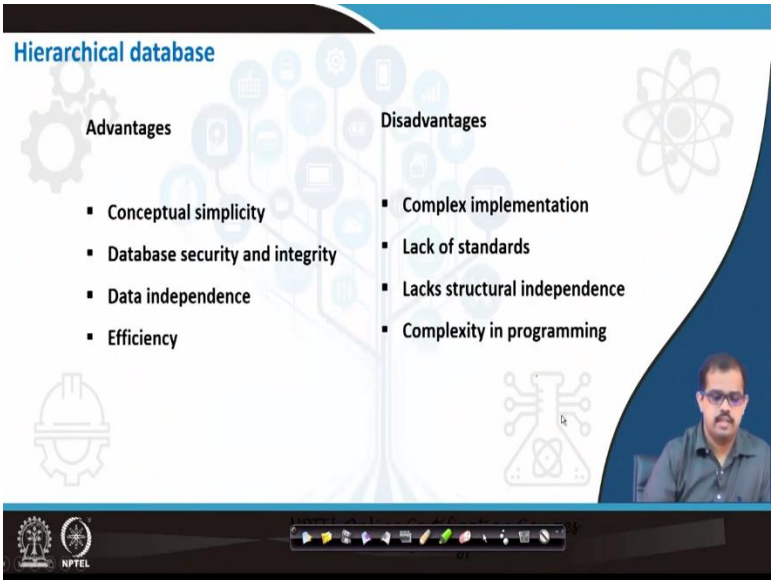
So these here when you see this is representing a standalone. So you have a particular database you have a query mechanism and that is how it is being accessed. So you are again have another database another query another database another query. So but there is a giant network which is

combination of different entities, different relationship in that entire database and you have different users accessing it for different purposes that normally we discuss.

There is a joint network wherein you have one giant network and multiple networks that are connected to this particular network and each of this network having a the maybe it maybe a standalone network or again it maybe a another giant network having a multiple users this come a joint network. So combination of standalone network, giant network or many giant networks form a joint network.

So joint database design provides the greatest benefit, but it is most complex in design in terms of designing and maintaining of this database is also extremely difficult. And most importantly security of the joint network is the aspect that one has to understand in order to maintain this entire database ok so this this is what about the hierarchical database.

(Refer Slide Time 05:21)

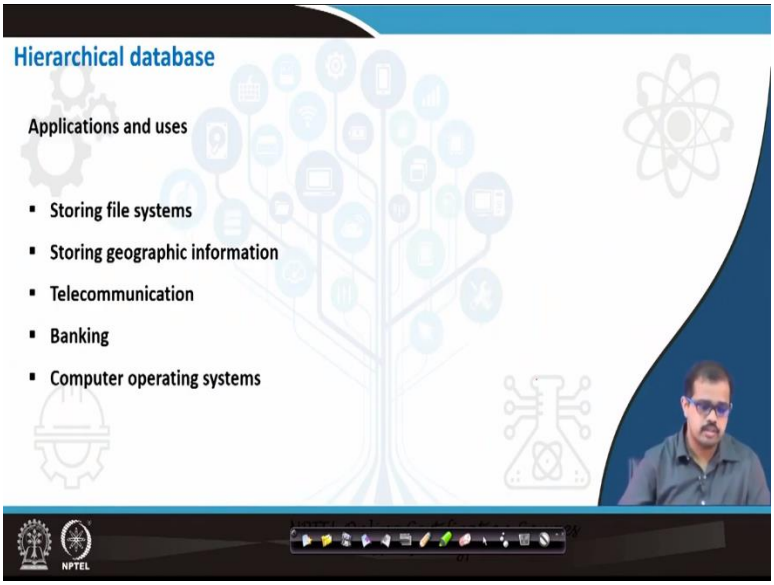


And similarly when we look at the advantages and disadvantage when we look at advantages it is conceptually it has simplicity ok. But when you look at the implementation part of it, it is highly complex especially when you have a joint kind of hierarchical database. And when you are looking at database security and integrity it is simple terms of implementation but when you look at the standards it lack standards in terms of whatever the data standards it has to represent or the data modeling standards that it normally has to represent ok.

So when you look at data independence the main advantage of hierarchical database that normally seen with database in advantage it has data independence. But it lacks structural independence for example it cannot have its own way of representation of the data. So structural integrity is missing so the independence is missing in terms of hierarchical database. And efficiency is actually missing for example when you have huge joint network it may have the efficiency of that particular system is high but the it is highly complex in terms of programming. Efficiency is good but it is highly complex as I said previously in term of joint.

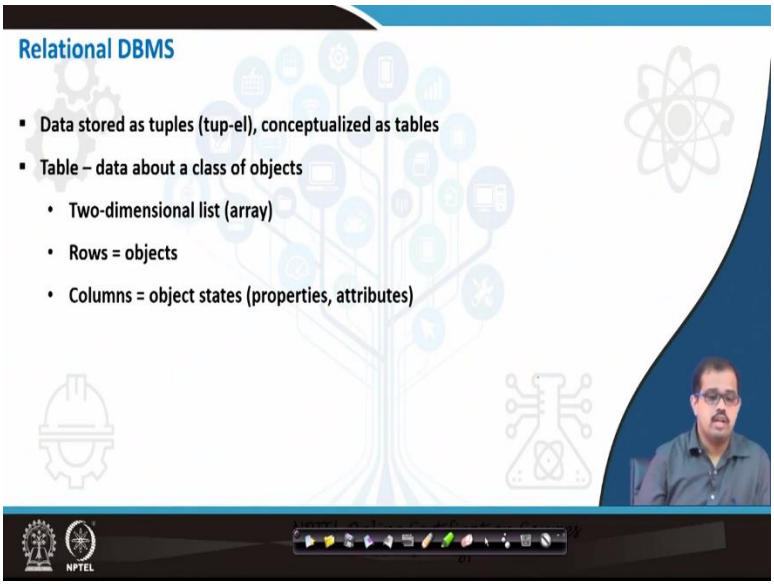
Otherwise if it is a normal standalone network it is much very easy in terms of both efficiency and complexity. But these systems are not preferred when you are accessing a different database at different point of time. So you need have either a giant network or a joint network which becomes complex in terms of programming.

(Refer Slide time 07:08)



And when we look at application and uses in hierarchical databases is normally are useful in storing file systems in a computer. It is storing a geographical information, telecommunication, banking, computer operating systems all of these are once used but now most of the system have gradually graduated into relational database management system than hierarchical database or they have gone into relational databases usage which has improved.

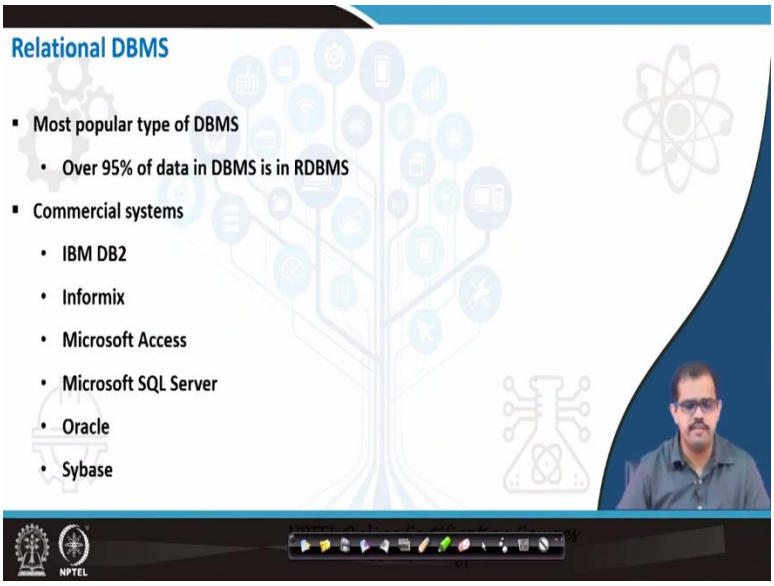
(Refer Slide time 07:41)



So when we look at relational database management system this is one of the very well-used database management system. Most of the company's worldwide well-known companies used relational database management system. Where GIS are whether it has any other company related to working on a database or developing a database most of the companies use a relational database management system. Normally a data is stored in a form of a tuples when I say tuple or tuple, so it is conceptualized as table's ok number of rows and column.

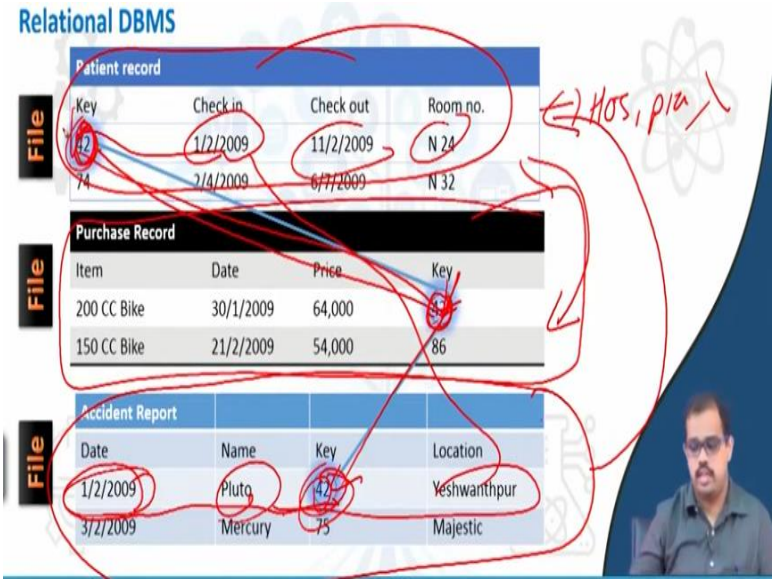
Columns are representing the fields. So each table data is about a class of objects okay. When I say objects it is two dimensional list rows is nothing but the objects and columns is objects states that is a properties or attributes that is called as fields ok so please remember this. Objects is when you have objects and the fields of that particular object then it is called tuple. So each of these rows is representing an object and each of this column is represented by a fields ok. So that is representing attributes.

(Refer Slide Time 08:57)



So when we look at DBMS as I said 95% of the data is in database management system is using the relational database management system. And when we look at the commercial part of it we have IBM DB2, we have Informix, we have Microsoft access which is again based on the relational database management system framework, then you have Microsoft SQL server, oracle, Sybase. So all of these are examples of using a commercial system for a relational database management system.

(Refer Slide Time 09:33)



To give you an example of an database for example I have a primary key which is 42 okay this is a patient record in a hospital. Let us say we have a hospital X where we are trying to access this patient record. So 42 is the key of that particular patient or a unique ID for that particular patient.

The same unique ID for example for anyone for a is not from the hospital he or she want to look at when the patient has come and where he has visited for him this particular file is quite enough ok.

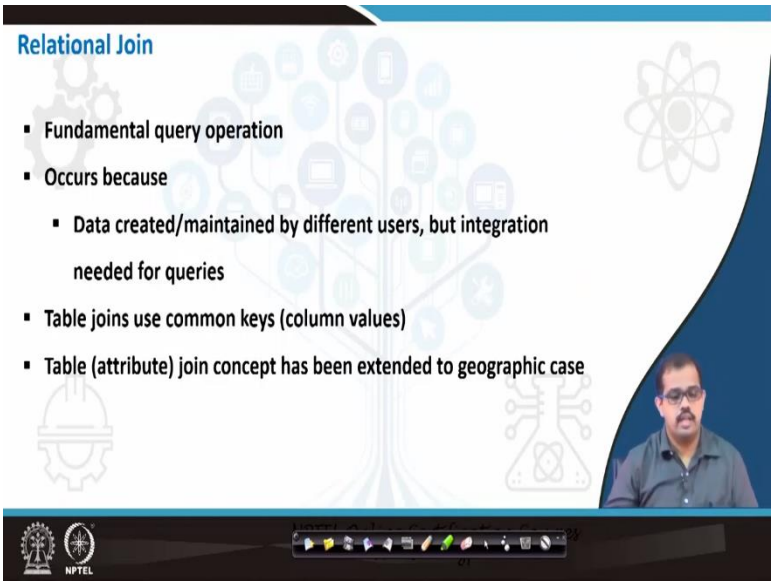
So if that particular person access this database is so he can see the patient 42 has checked in at this point of time and checked out at this point of time and this was his room number ok so about 10 days. Now if let say a management person who is managing that particular hospital ok or particular system has to look at what patient what is the patient has consumed? How much the consumables are required for that patient surgery or any of those aspects and what is the total price?

So he or she will be gaining access to this particular table in this same database. The relationship is built with this particular key. So this is a primary key in this particular table and the same primary key is used as a key here it becomes a foreign key ok. So but the relationship is built with 42. Now there is an administrator or let say police who wants to know the entire history of this particular patients. He or she may not need what is the purchase record of this particular patient ok. So he or she only needs a where the accident has happen?

How and what is the status of this patient where it has checked in and checked out. So this for example a police inspector may get an access to this particular table along with this particular table ok. Whereas a manager in the hospital who is looking at managing records may get an access to these two things ok. So this particular the police look at the same key here which is 42 so if he puts in the patient name and he know the key so he is defining this particular key and because of which he know when the accident has happen?

What is the name of that particular person and where it was where it is a location and also when he was admitted to the hospital? When he was recovered from that particular injury and where he recovered? Okay so he know the entire database of it. So this is what is a relational database. It is relation that is built based on keys that is represented in each of this system. So that just to give you an example of what relational database is? I tried to build up for this particular example ok. So if you have understood this let us move on.

(Refer Slide Time 12:58)



Then there is concept of relational join when I say relational join it is fundamental query operation. So you are querying in terms of joining. When I say relational join it is fundamental query operation. So you are querying in terms of joining. When I for example let us simply think about join. What is join mean? If I have to join two things which means that I am trying to built a relationship between these two things right.

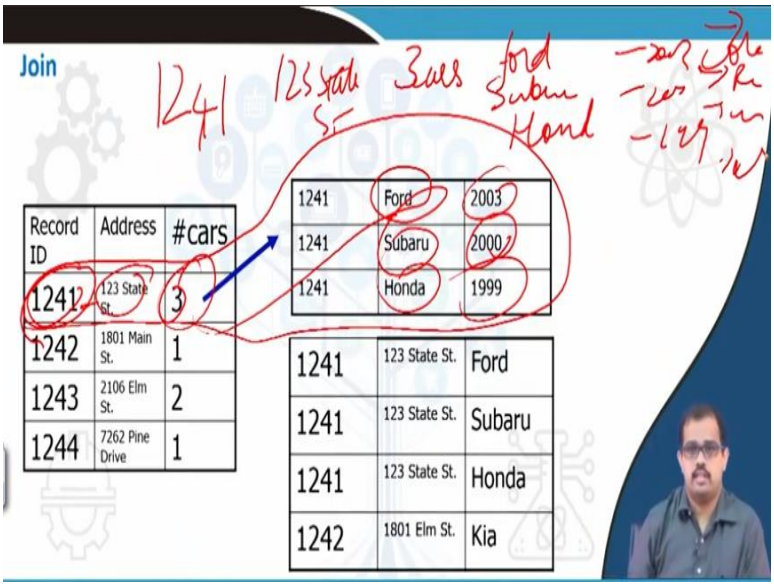
So that means to say that I am querying where I have to join basically whether I have to join at the end, whether I have to join at a particular point. Now which means that I am trying to querying a particular entity right. So it may occurs because your data created maintain by different user but integration it needs to be queried ok or for example you have common keys in different table. You want to merge all those tables to get a common output.

For example instead of policeman if I want to give him an output instead of giving him 2 3 tables at 2 different places I would give him the entire like a cart whatever he is able to access in a single table. So just use this process of relational join combine both of those tables which are represented at first or the third table and that can be represented as an output for that particular inspector or the policeman who is actually looking at it.

So that is what it is a use of a relational join. So table attribute join concept has been extended even in the geographic case. If you are looking at the geographic case if you want to join 2

different aspects of that geometries that are available in the spatial use. So that also can be used in terms of a spatial join spatial relational join if you are trying to use a relational database.

(Refer Slide Time 14:48)



For example let me give an again a example here now I have different addresses of different people for example I have a record number here. 1241 is a record number then I have address that is corresponding to this particular record number and there are different cars. For example a person with 1241 record has 3 cars, a person with 1242 record has 1 car, a person with 1243 record has 2 cars, whereas a person with 1244 record has 1 car.

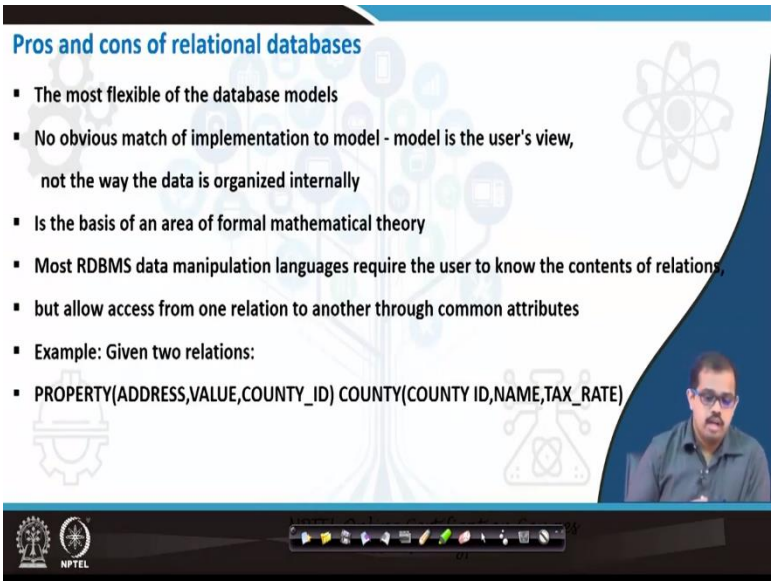
So each of these records have different cars but the very important point is this is represented in form of a geographical quantity which means each of them have a address that is actually representing a geographical unit on the earth surface. So this is record this is the address and this is a car. Now if a person wants to know which are the cars he has from which company and when he bought. So probably using this particular record ID he or she can access the number of cars. The car brand, brand cars and when he or she has brought this particular car right.

So now if I have to represent for example I make a query saying that 3 cars of 1241 address and the date of buying for example if someone wants to query about that particular thing. So just with a query if the joint will join both of this tables ok and it can it would represent something like this 1241 ok address is 123 state street 3 cars ford Subaru and Honda ok bought at 2003, 2000,

1999 ok. If again someone gives a query to ask color, if the color is there in other table so that is also join into it as red, green and black.

So relationship relational join tries to give information by joining different tables and combining the information and giving a as a planer information into or a flat file information to the user who want to access such information's. That is was that is what it is a use of join.

(Refer Slide Time 17:29)



So when we look at pros and cons of relational join it is the most flexible as far as the database models are concern. No obvious match of implementation of a particular model is the users view not the way the data is organized ok. So whenever I am speaking about the data models in a database how the user view has to be there. It is not about the data ok. Data organization is based on the normalizations and the database management system and not with the models ok.

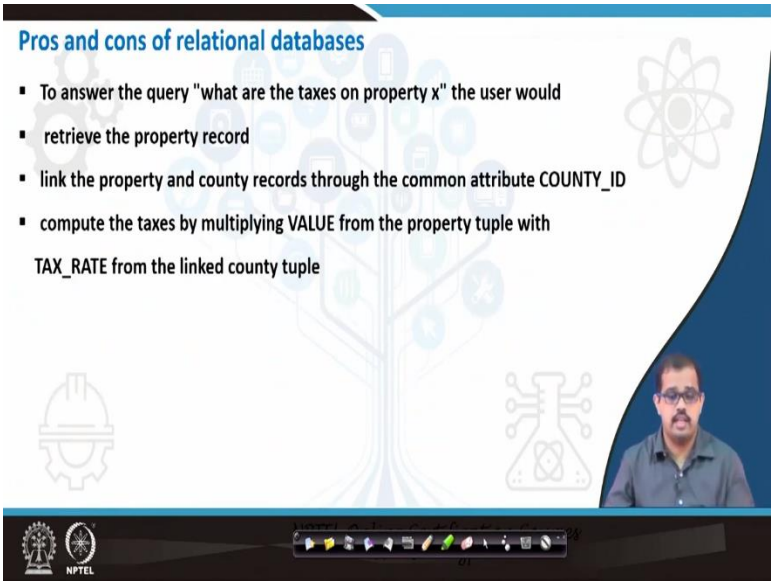
Is the basis of area of formal mathematical theory. Most RDBMS data manipulation languages require the user to know the content of the relations. So that is what is very important if you do not know the relation what kind of the different relations exist and where it is exist probably that querying may not be very effective ok. It allows access from one relation to another through common attributes. So the common attributes has to be there in all the different tables.

If that common attribute is missing, then your query will miss a particular table in the context. For example the property, the address value, ID country and the and the same country will have

country ID. So the country ID is a common here so you have a name and tax rate. So if a person who wants to know a property details he or she would be given a access to only this. If someone wants to know the country details, he or she where the property is located he or she would know only this.

But if a person wants to know details of the country where the property is located the details of the property then both of this table can be related using this particular key that is common in both of this tables. One at one place it is primary key and another place it is a foreign key. So based on this you try to join that table and provide as a output. So that is how relational database works it is a simpler way of putting it but when you look at the entire organization database it is quite complex in terms of maintenance.

(Refer Slide Time 19:34)

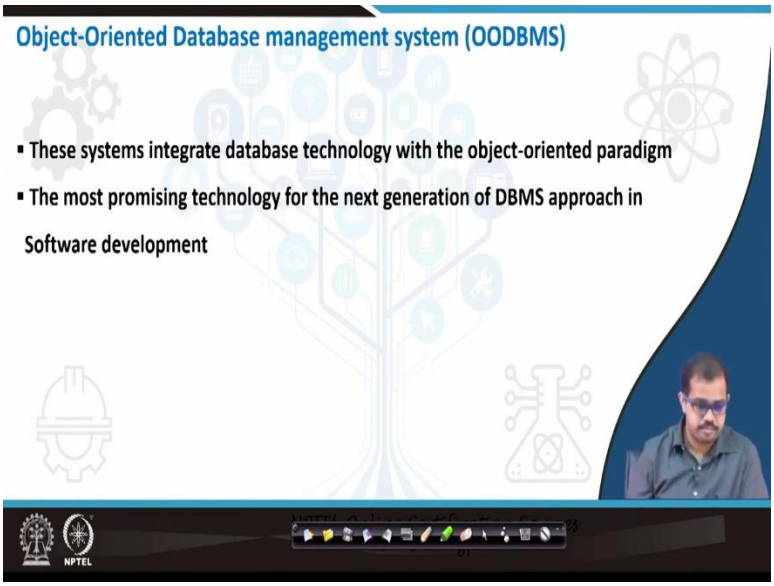


So when we look at the pros and cons to answer the query what are the taxes on property x the user would retrieve the property record first of all and link the property and county records through the common attribute county ID ok. If compute the tax by multiplying the value from the property tuple with tax rate from the link county tuple. So that is what user has to do? It does not give you the entire output as just if you want to compute tax for that particular property in the county.

So it does not give you an output but it gives to the entire table where you have to probably use data manipulation languages or add in attribute information into that particular table ok. So that

is the pro of it is you can easily look at the relations and join and query whatever you need. But the con is that the user that has to look at all of these aspects.

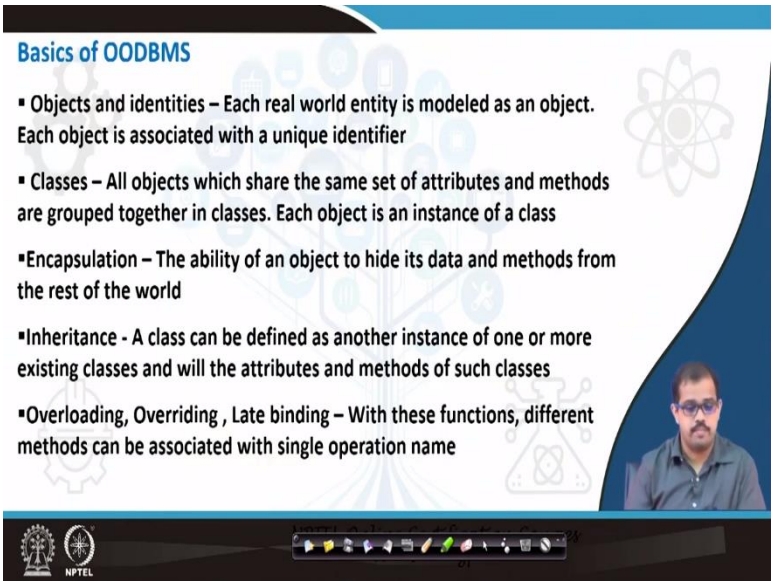
(Refer Slide Time 20:43)



In the next type of database management system is object oriented database management system. These system integrate database technology with the object oriented paradigm ok. The most important promising a technology of next generation of database management approach is in the software development. Object oriented is one where every object is treated as a different entity ok.

Each entity has a different table and each wherever you have an object is connected through a relation and each of this relations can be modified. That is again a object relationship database management system that is also another kind of a database management system.

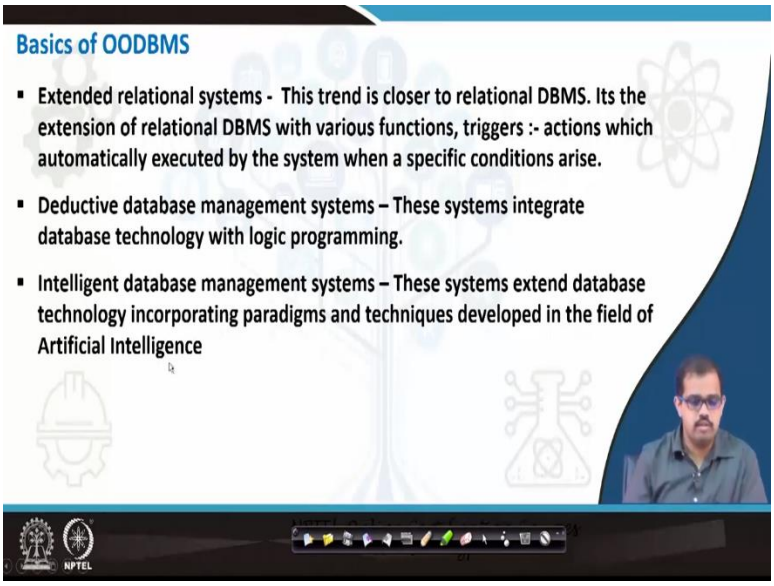
(Refer Slide Time 21:26)



So when we look at object oriented database we have objects and identities each real world entities is modeled as an object. Each entity is modeled as an object and each object is associated with the unique identifier that is very important. If this it always has a unique identifier and you have as an object oriented programming you have classes here. All objects which share the same set of attributes and methods are grouped together in classes ok.

Each object is an instance of a class ok. Then the ability of an object to hide its data and methods from the rest of the world is done using encapsulation in object-oriented database management system. It has a property of inheritance which means it class can be defined as another instance of one or more existing class with attributes and methods of such classes ok. Overloading, overriding, late binding so these functions different methods can be associated with the same operation name ok in terms of object-oriented databases which is not possible in relational database as such.

(Refer Slide Time 22:48)

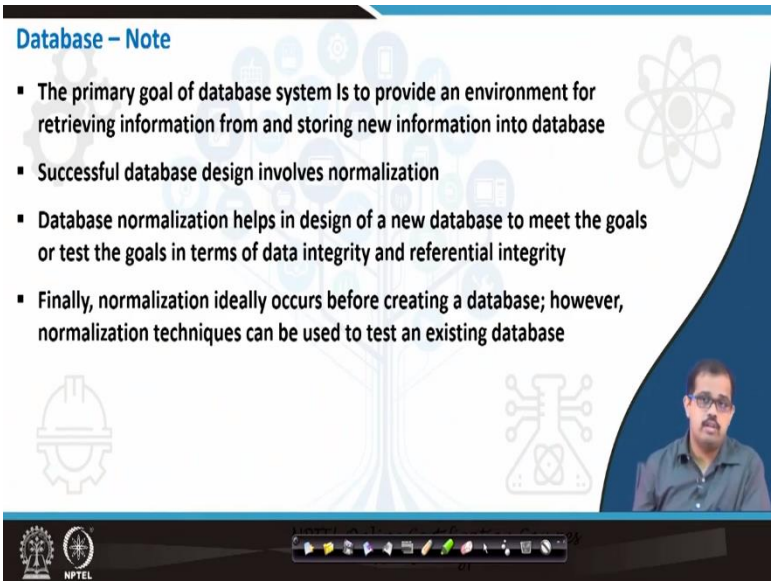


And when you look at extended relational systems is trend in closer to relational databases that is object oriented relational database management systems these are extension of relational database management system and with various functions triggers, actions that are automatically executed by a system when specific condition arise which was an disadvantage in terms of a relational database.

It is also a deductive database management system these systems integrate a database technology from with a logical programming. If you have a logical programming or a logic program into it so it gives you a detective database management system. And also provides an intelligent database today order of today's days is providing a intelligent database management system that it can provide it can incorporate an artificial intelligence into it.

It can have incorporate different various field of where the technologies imported in terms of a database management system or managing the entire database providing the queries or to the user in much useful way in terms of user manipulation or user usage is reduced. But you get a query much in a much easier way. Such things can be done using a intelligent database management system in terms of object oriented database management system or an object oriented relational database management system.

(Refer Slide Time 24:15)

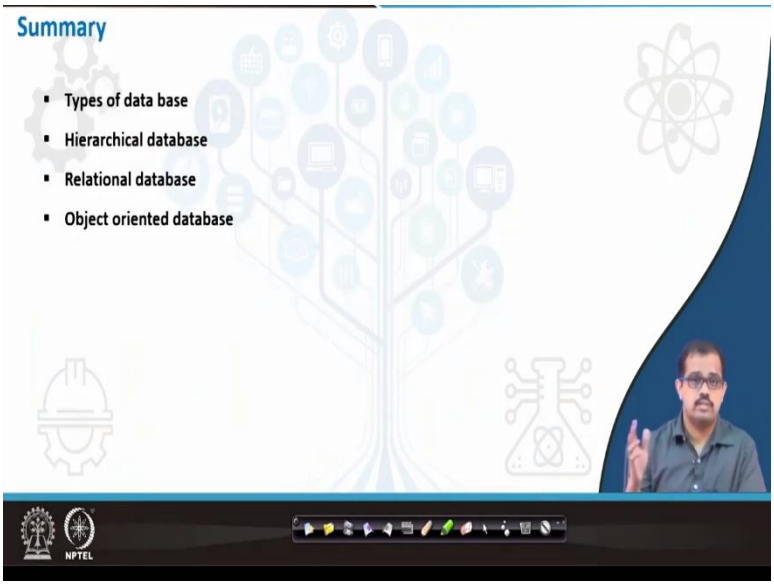


On a note when we look at database the primary goal to pull it up together the primary goal of a database system is to provide an environment for retrieving information from and storing new information in the database ok. Successful database design involves normalization basically. So different types of normalization also we will look at. Database normalization helps in design of a new database to meet the goals or test the goals in terms of data integrity and referential integrity. So keep these in mind.

So data integrity and referential integrity are very important. Finally normalization ideally occurs before creating a database however normalization can be used to test the existing database. So this is about what we learn in the entire database system. In the next class we would be looking at what are different normalization techniques? What are different normal forms? ok. How do we start with the database?

How do we convert each database into normal forms and finally ended with the best normal form where in the database can be easily and it is much quicker and most importantly it is more efficient in terms of handling?

(Refer Slide Time 25:34)



So to summarize todays class so we look at the types of databases, we first look at what is the hierarchical database? We looked at what are the pros and cons of the particular database. Then we looked at relational database how a relational database is advantages in terms of all the database queries. We looked at pros and cons also we looked at the cons the major cons is that it cannot give you the final output that you would expect.

But you would give you the entire data where in the output has to be manipulated by you as an user and we looked at object oriented database, object oriented relational database where in object the entire entity is converted as an object and it has different ways of access and usage of including artificial intelligent techniques or intelligent database management system in order to provide outputs to the user. So this about a different data models.

So in the next class we will look out normalization as I said previously different normal forms and probably end this database session with giving you some examples of that. Thank you very much till have a nice day time until then.