Spatial Informatics
Prof. Soumya K. Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 17
Spatial Indexing – I

Hello; so we will continue our discussion on Spatial Informatics. Today we will be discussing on Spatial Indexing rather, for next one or two lectures we will be dealing with this. This is one of the very fundamental aspects for any spatial data or more for spatial query resolution right. Rather for having efficient resolution or resolving spatial queries we will see why how and type of things. Just to little go back to our other means our previous discussion.

So, what we have seen spatial data are pretty large in size. So, it has both IO and computational cost is involved in the process right. So, this so in order to make this efficiency, efficiently querying the things, what we have seen in our previous discussion that we have different query optimization techniques. Like, there is a general tendency of bringing down these non-spatial portion of the queries, below the execution tree so, that the number of tuples for the spatial predicates reduces.

And so, the overall cost of querying reduces right so, this is one and of the things. We have based on the different cause base metrices we can also reorganize the things etcetera right. Or even we have seen that type of things that based on the number of tuples we have a metadata informations we can look into the query in a things. Nevertheless, we can have different query execution tree, but which has the same type of outcome each has to be the same type of outcome right.

Now, giving all those things in place right, we all this query optimization in place whether I can make something at the underlining data organization. So, that the querying will be faster right like for example, suppose I want to query something which is on the land base right. I want to find out something, which is on a particular region of the things right in a particular region of interest.

Say a ports in something within IIT Kharagpur campus right, I want to I am querying on some something on land use land cover. Now, in order to do that I need to optimize the

query the way things goes on, but end of the day your data is in the disk right. So, the disk will be stored as per the disk management policy, not based on your querying policy per se right.

So, in other sense the if you can think that the this the data is stored in some fashion. Now, in order to query on the for say query which in a particular region say IIT Kharagpur within the campus, now the some data will be here then the discourse for a skipping some portion extract the data there. And, go on like this right pretty fine the answer will be optimization strategy is also in place.

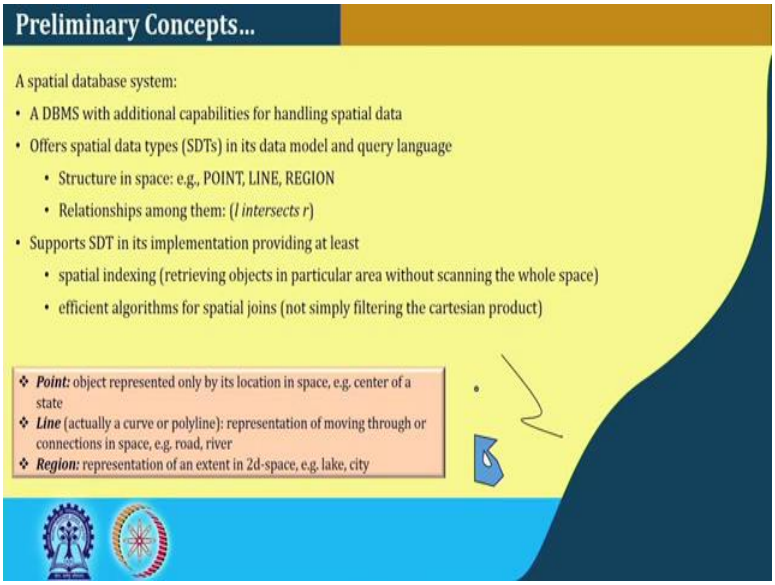
But, had it been I have I can if I could have kept this data, within a particular region then my query could have been faster; that means, the disk movement I could have reduced right. So, or in other sense if my indexing instead of 1, 2, 3, 4 that way if my indexing is some other fashion or then we could have we could have a better efficiency on accessing the data from the disk right. In other sense we overall the efficiency will be increased right.

So, in order to handle those type of things which is important in indexing. So, it is not only for spatial, indexing for any data base is one of the important aspects right, we want to like to see the spatial extension of the things. So, though it is not that will be we will be thinking that you know everything on the on indexing per se. But in general that what is the concept of indexing what we mean to do look at right.

(Refer Slide Time: 04:34)

So overall we want to look at this two aspects spatial indexing concept wise and the type of spatial indexing out here for what we will be using in this our spatial context.

(Refer Slide Time: 04:47)



Now, some before going to the next let us take some quick recap of this our some basic concepts or which is which I should say which basically drives out for this spatial indexing. So, what we have seen this spatial database management system. So, a DBMS with additional capability of handling spatial data correct. So, it is a fundamentally a DBMS with the additional capability of things offers spatial data type SDT's in its data model and query language.

So, structure in space that is three things are important like point, line, region or point, polyline, polygon type of region relationship among them right. Like line intersects r or polygon a region r one overlap region r 2 so, these are different type of relationship between them which are not traditionally our in our standard databases. Suppose SDT or in its implementation providing at least spatial indexing.

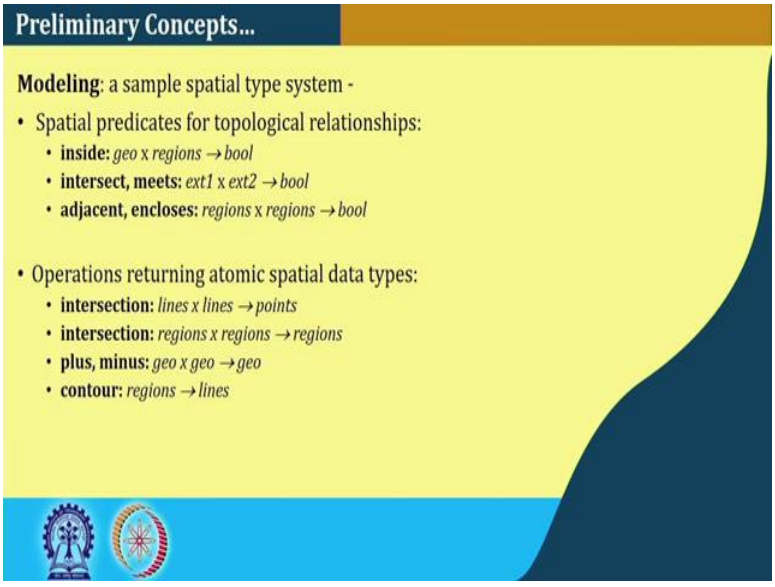
So, we require a spatial database system which will suppose these a the spatial data types in its implementation over at least like using spatial indexing retrieving objects in a particular area without scanning the whole space. The idea is that the indexing is such a way so, that I do not have to search the whole space and get these data out of it. So, that I can basically my search space should be reduced so, that I overall efficiency efficient algorithm for spatial joins.

As we have seen this join is a costly affair or spatial join is a very costly affair so, we need a efficient algorithm, so spatial join not simply filtering the Cartesian product. So, or as I say we just try to filter the filter out the Cartesian product of our non-spatial or traditional data set we require some efficient algorithm for spatial joins so these are may be the type of things we require.

Now, again just we have three type of object point line or polyline region or polygon right. So, point object represented by one location in space like this is may be this is a point, line curve or polyline represented moving through or connecting space by a route river. And, region representation of the extent of the two d space like lake, city some any which has a polygon feature right.

So, these are the three fundamental feature sight and anything we can do with this combination of these things right. So, when we look about indexing, we also keep in this mind that these are the things which are need to be looked into.

(Refer Slide Time: 07:37)



Now, when we look about modeling like a spatial type or datatype systems we have a spatial predicates for topological relationship. Like insight so that is one geometry geocross region to a Boolean. So, I want to find out that whether it is inside or not to and for intersect and or meet type of relationship ext 1 cross ext 2 also Boolean.

When we say that this re this region intersecting of these or this line intersecting with the these or anything that will be in a Boolean thing adjacent region cross region also Boolean right. So, it is adjacent or encloses and type of things. So, I have a definition of adjacent so that I take two regions and their relationship will be whether it is adjacent or not.

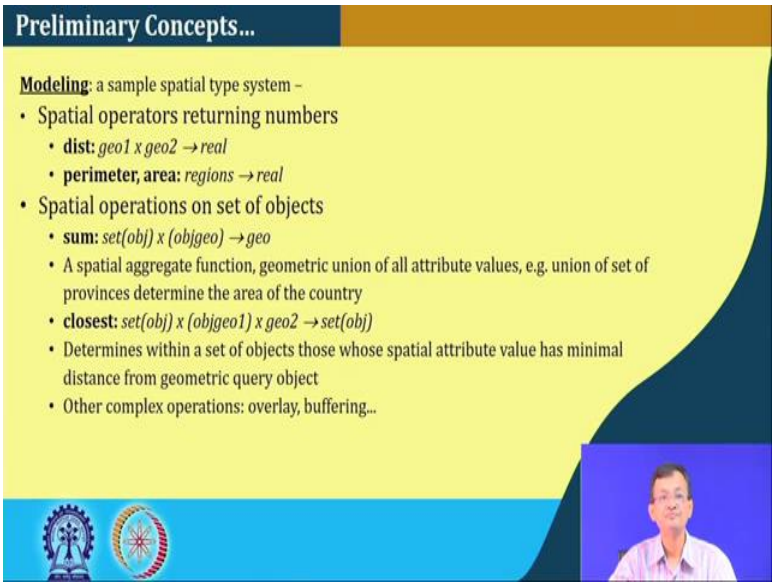
And we have different other type of operations like some returning atomic spatial data types like intersection line cross line is intersection the outcome is point. Or points if it is a polyline and they are intersecting there is a point set of points. Inter section with regions cross region will be a region like two region intersects the things will be a region plus minus so, two geometry will return to a another geometry.

So, if you have operations of plus minus, contour we will have a region and the contour will be like right. So, what we try to see here there are different type of spatial predicates there are different type of the spatial operations, which has different outcome or when they when they are applied on this sort of things like binary or unary type of operations.

So, again when we look out the indexing we need to keep in mind that this type of operation is coming into play right. There will be a query which intersection comes into their there will be a query where there may be inside operation and so we need to cater to this. So, that my indexing will be such a way that it is efficient right. So, there may be different types of indexing, which may be supporting different category operation much in a better way and type of things that is one part.

Secondly, I may sometimes have a priori notion of that what sort of operations I am expecting right not always right. So, it is not like that everything very random right. So, I have some queries which I have a priori notion that which are coming to that, in that case I have a indexing two place. Otherwise, I can have index i j k different type of index I pull when I require those.

(Refer Slide Time: 10:48)



So, modeling a sample spatial type system so spatial operation returning a numbers. So, so it is one is that there are there are so continue with the like distance two geometry returns a real like I say what is the distance between two points right. Or distance between a point and a region I may have a definition right. So, I have a point to the region it will be the centroid of the region into this point that may be the thing.

Or the nearest boundary point of the region that may be a thing two line distance between two line. So, there should be a definition of the thing so I have two geometry classes and then I can have a spatial operators called distance and then I that that returns a real value perimeter area. So, perimeter area typically for some region or polygon which has a again a real number to do right.

So, there are other spatial operations like sum set object, cross, object geometry to a geometry. So, I can basically sum up two regions to make something. A spatial aggregation function, geometric union of all attribute, that union of set of provinces to determine the area of a country right so, that there is a thing. So, I have say country of this number of provinces or states. Now, I want to find out that area of the whole country that is one thing. I want to find out the area so, I have a union of those things.

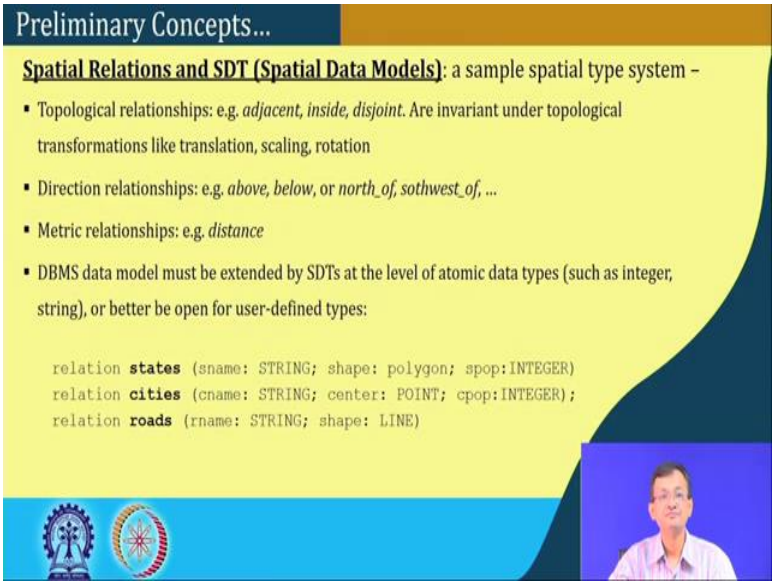
So, there can be a different way of handling those things right two regions I add and find out the perimeter. They there may be some geometric operations as to be done computer science geometric operation before going there closest set object, cross object, geometry

1 cross geo 2 to set object right. Determine within a set of objects those whose spatial attribute value has a minimal distance from the geometric of the query object.

So, I can I want to do a from a set of object some predicate or some condition to be satisfied like minimum minimal distance from that from a geometry objects right. So, I can have number of schools then I want to find out that the a particular hospital which is closest to this one of the schools right. I can have different type of region villages etcetera then I during flood I have a rehab center type find out that which is the or I have a number of rehab center say depending on a village which is the nearest things etcetera.

So, from a set of objects, I want to satisfy some condition other complex operation like overlay buffering things are there right. So, again what is our notion as this are they are in the spatial query set. I need to setup by indexing in a such a way so that my data retrieval will be efficient in a first order manner right so, that is one of the things etcetera ok.

(Refer Slide Time: 14:00)



So, what we are trying to see that what is our basic background or motivation or going for indexing. Again let me repeat that as the data are pretty large right. So, if when we store on a disk etcetera, they are sprayed over large displace not only they are in number of cases it cannot be it may not be in the one cluster it can be in a multiple cluster right.

So, accessing those things or always becomes a costly affair right or often become a costly affair. So, had it been my based on my type of operations which my I have appropriate indexing structure where it the data which are need to be accessed are closer. Or within the particular disk cluster or a particular disk then my access rate could be much faster right.

So, overall efficiency in dissolving the query may be first of all either right. So, spatial relation and spatial data models or spatial data types; so, to say a things like topological there are topological relations. Like adjacent inside disjoint these are the topological relations right; are invariant under topological transformation like translations, scaling, rotation and type of things right.

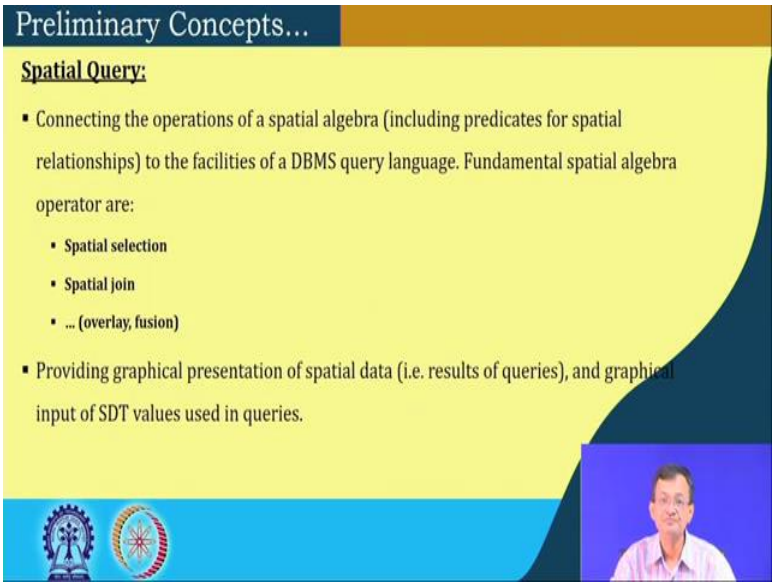
The adjacents will be always adjacent, if two objects are adjacent they are adjacent whatever that topologically whatever that topological transformation into. Like one object inside another objects or one object not touching or disjoint another objects will be always disjoint, whatever the map we want to this format type of things.

Direction relationship above below north of south of these are different direction relationships are there. There are metric relationship like distances we have seen different metric distance metric area and other type of different type of metrics will be there. DBMS data model must extended by SDTs at the level of atomic data types like such as integer string or better be open for user defined data types right.

Either I have to extend it to our atomic data types or some other user defined data types will be there. Like relation state may have a state name which is a STRING; shape which is a polygon and a say population which is a INTEGER right. So, or a city will have a name STRING population and a city center which is a POINT time right and similarly road have a name and a say which is a polyline or LINE type.

Now these point line polygon these are a geometric properties right. So, geometric properties these are either it has to be in that fundamentally, in what you know that integer float or real and type of things as to be boiled down. Or we need to cater this as a single entity so, somewhere that DBMS has to support them.

(Refer Slide Time: 17:21)

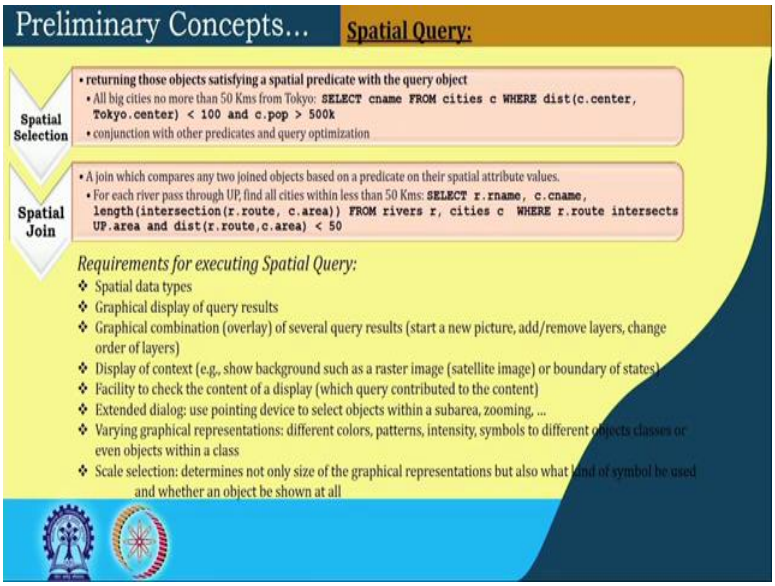


Other for all whatever we are trying to do or whatever we are trying we are discussing that one of the major implication is to have a query resolution right. We should be able to dissolve the query in a efficient manner. So, connecting the spatial operation of a spatial algebra or relational algebra what you have seen like, including predicate for the spatial relationship to the facilities of a DBMS query language fundamental spatial algebra operations are this right.

So, what we are trying that connect that whatever the operations we are of a spatial algebra, has to be connected to this fundamental which is supported by the DBMS right somewhere or other right. So, that which is that query resolution can go on. Then what are the fundamental operations what we are having spatial selection, spatial join, overlay, data fusion and type of things are some of the things we are having. So, providing graphical representation of the spatial data, that is the results of query and graphical input of the SDT types used in the queries and type of things right.

So, it also provide that so that we can have better visually handle the query with visual interpretation and type of things.

(Refer Slide Time: 18:50)



Now, just to we have gone through spatial queries in a bigger way. Just to summarize and again try to find out that where our the need of indexing or motivation for spatial indexing into the things right. So, spatial selection it returns those objects satisfying a spatial predicate with query objects. Like we are doing that all big cities no more than 50 kilometer from Tokyo select c name from cities c, where distance c city center Tokyo city center of the Tokyo is less than 100 and cpp greater than things it should be less than 50.

So, there is a typo that within 50 kilometer and something that population more than 5 the queries like all cities within 50 kilometer or the queries for within 50 kilometer and the population more than 50 k. Conjunction with other predicates and query optimization is possible, spatial join a join which compares any two joined objects based on the predicate on their spatial attribute values right. For each river passes through up find all cities which are less than 50 kilometer from that river right.

So, that is one thing is that how do you what is the distance from a city and a river like, if you consider river as a line object city as a polygon. So, the city center may be the your centroid of the thing and then we calculate the stuff right. So, the query if you see select r name river name city name length intersection between r route city area from river r city I where r route intersects with the up dot area. And type of things see there are couple of things which is important out here we want in up Uttar Pradesh a city in a sorry state of

India. And then want to find out all cities which are less than 50 kilometer right. So, each river passes through up.

So, first of all I need to find out if the river route is within the UP and cities which are 50 kilometer from that particular river right. So, that query may be not making much sense, but nevertheless if these type of queries are there so, I have a more complex type of representation of this SQL right. And what is interesting to see that there are different type of operations come into picture.

Like, there is a intersection operation which is a spatial operations, where as there are length operations based on this intersection, we want to find out the length between this particular intersection thing. And the from that particular city to that particular river that what is the length of the thing from the river and cities where r route intersects UP area and distance r route and city area is less than 50 right.

So, there are lot of spatial operations coming into play with the data sets which are a route river data set. And there are data set from the city data set and type of things. Now here also one is that the data is come the data is ready for the query to be solved other way if you if you if you look little deeper it has to follow back to this backbone data sets to get into the data.

Now, the data is distributed over large number of spaces, which may be may so happen some of the things are (Refer Time: 23:03) able. Like route data may be with transport authority and this administrative data may be with state authority or something some other authority. And then it becomes difficult to bring them together and make a meta data so a priori I can we can always do that.

Otherwise even when we are accessing the city like say the river data may be having the whole dataset of all India basis right. Had it been appropriately indexing on a state basis the access rate would have been much faster right for at least for this query right. So, that that again that in that they are the indexing things again come into play where, I have a better access using appropriate indexing of the same data set right.

So, requirement for execution of the spatial query if we see so, first of all it has to handle spatial data sets that is one fundamental. Graphical display of the spatial query at times become important because, representation wise. Graphical combination overlay of

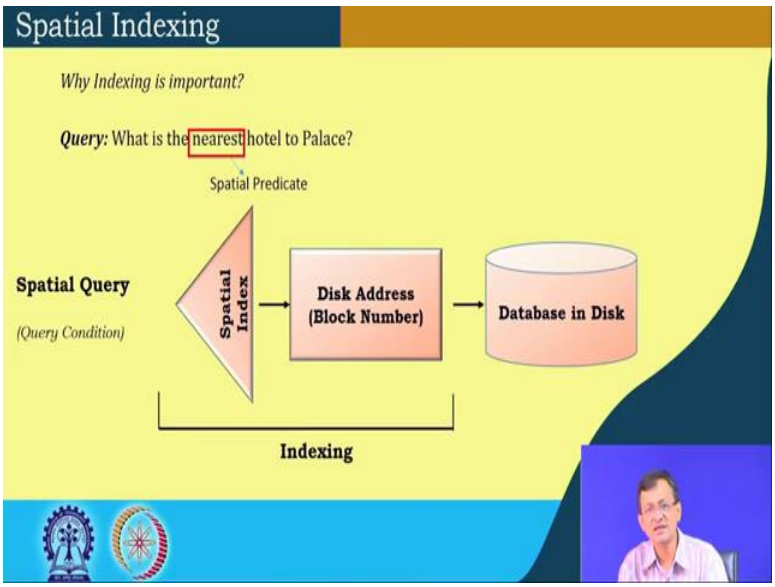
several query results, start of a new picture add remove layers change order etcetera things are there.

So, graphical combination overlay type of things display of context background such as raster image satellite image boundary of the state display of the context. Facility to check content of the display extending dialogue use pointing device to select objects within the subarea and zooming it, right. Varying graphical representation different color, patterns, intensity, symbol to different object classes and so and so forth; scale selection determines not only size of the graphical representation, but also it looks at the what are the skill and type of things right.

So, it is also important in spatial context the at which scale it operates. First of all when you combine the scale of the two things will be same. So, if you remember we discussed the scale is basically at the what you represent and what the reality in the ground what is the ratio of the thing. So, when we say one is to fifty 50,000 scale; that means, 1 meter on your paper on the map is 50 meters on the 50,000 meter on the ground. So, it is representing one is to those ratio right.

So, size of the graphical represent, but also the kind of symbol to be used and whether the objects could be shown at all etcetera all those becomes. Because, at a when your map becomes a larger scale then, you may not see some of the objects then what sort of symbol is etcetera etcetera that is important.

(Refer Slide Time: 25:56)

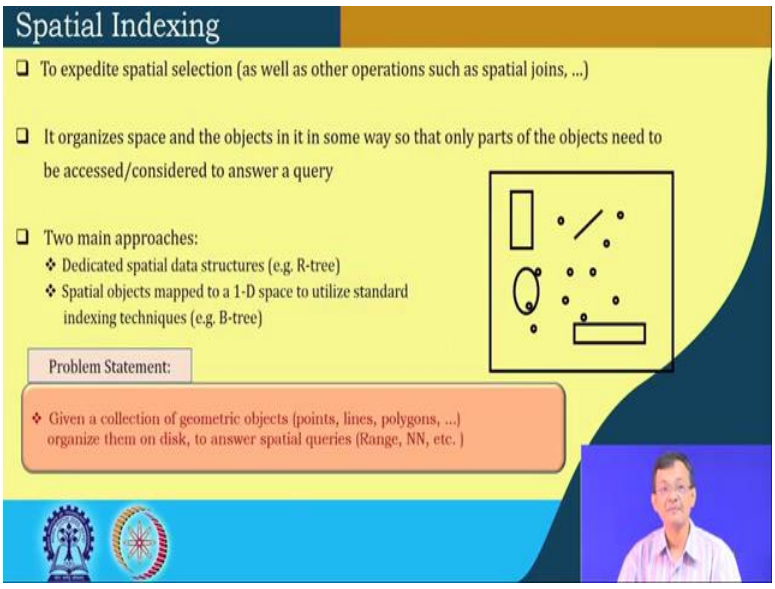


So, all these things lead to a need for this spatial indexing right. So, why indexing is important midness to save we have we discussed a lot on that things. So, like what we have need to find out that what is the nearest hotel to a particular palace or a particular monument or particular point of interest right. So, this is a predicate. So, we need to find out that nearest things from churning the data like so that is my query.

So, SQL representation at the thing it has to search the data set. So, there is a hotel data set there is a place palace data sets which need to assign and find out the distance between them right what is the distance. So, now, we need to find out the given a particular palace what is the nearest distance. So, if we look at the spatial query that query condition, it requires appropriate spatial indexing so that the search will be efficient. And the based on the spatial indexing, I need to look into the or disk addressing as to be there.

Or reorganization of the disk addressing is required into the spatial in a in a back end disk management thing. Based on this disk address or block number the underlining database need to be accessed. So, though query is the inducer requirement or our requirement look at, but that appropriate spatial indexing or good spatial indexing with a proper synchronization between the this whole synchronization; with the whole system allow me to access the data in a much faster way.

(Refer Slide Time: 27:50)



So, to expedite spatial selection as well as other operations such as spatial joins etcetera its important. It organizes space and object in it in some way so that only the parts of the objects need to be accessed and considered the considered to address or considered to answer or resolve a query. Like I say that I may having 1 billion data set right of having this say river, city or different cities etcetera. Now my interest is finding out a city cities within a particular country right.

So, if the cities are appropriately indexed based on the country wise then I can access this portion of the things that is why I am telling my search space and the object only part of the object need to be accessed. So, long my countries India I am accessing, but if it is distributed with a 1 billion data set then accessing will be difficult, but otherwise within that particular thing right.

So, there are we can look at two major approaches one is dedicated spatial data structure that is R tree etcetera your we are conversion with t tree structure for searching something. Like the what we have seen that in our traditional or standard DBMS the data we access is to B tree or B plus tree. Like spatial object map to 1 dimensional space to utilize standard indexing scheme that is B plus tree so, that can be one way of looking at it. .

So, so problem so if you look at the overall the problem at hand or that what needs to be solved or at least address by the indexing is that given a collection of geometric objects point lines polygons. Organize name on disk to answer spatial queries right like range NN and type of things in a efficient manner right.

So, what we say given a collection of geometric objects like point type objects line type polyline type or polygon type objects; we want to organize them on a particular disk to answer spatial query in a efficient manner. So, that the overall means time complexity reduces or the overall query resolution time reduces.

(Refer Slide Time: 30:28)

So, there are several indexing operations like, operations on those structure like insert delete and query types also that. We have already seen that range query, nearest neighbour and distance can these are the standard types of points and type of things. And there are for rectangle we have intersection query and containment query ok. So, for today's discussion if we try to see what we tried to look at that given a spatial query spatial operation spatial predicates. Wow why indexing is important and needed right that we try to address at one part.

On the other hand we try to see that what is the different type of approach for will be there for this indexing right. So, rather what we will do in our subsequent lecture on spatial indexing, we will be looking into the different aspects of the things will we be looking at the space filling curves and R tree. So, fundamentals of R tree and how it helps us in resolving these spatial queries in a efficient manner. So, with this let us conclude our today's discussion.

Thank you.