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

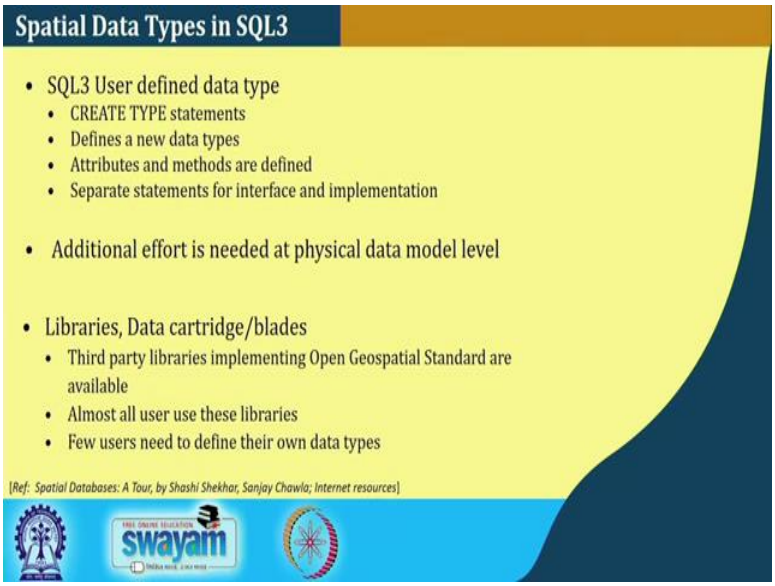
Lecture - 14
Spatial Query Processing / SQL (3)

Hello, so we were discussing about spatial query, so spatial query knows as spatial SQL in the context of our this course on spatial informatics. So, if you recollect last one or two lectures what we have seen that, why we require a special dealing for special data for SQL right. As we mentioned earlier that already we have a SQL in place right and that is very much standard and de facto language for any data base things why we require more than that. So, what we have seen definitely that is good for any type of query which has tables and others in a somewhat traditional tables and of non spatial structure.

But when you look at the spatial concepts like specially the topological concept, Euclidean concept those type of concepts. Then it is re we have to have a separate dealing like last discuss and we have we have seen that things like crossy cross, things like buffering, things like overlap those type of things like touch right. So, these things are is important like we have seeing that neighbor of a particular country, neighbors of India, so which are touching a though that country, so, which as a land mass as a touching into the that country.

So, the how do I define in a normal SQL, how do I define those type of a concept right. so those type of things are are dealt with in case of a SQL. Now, again what we have seen that, SQL 1 and 2 do not allow this standard do not allow this user defended. where as here we required different type of user data types vis a vis operations and type of things.

(Refer Slide Time: 02:23)

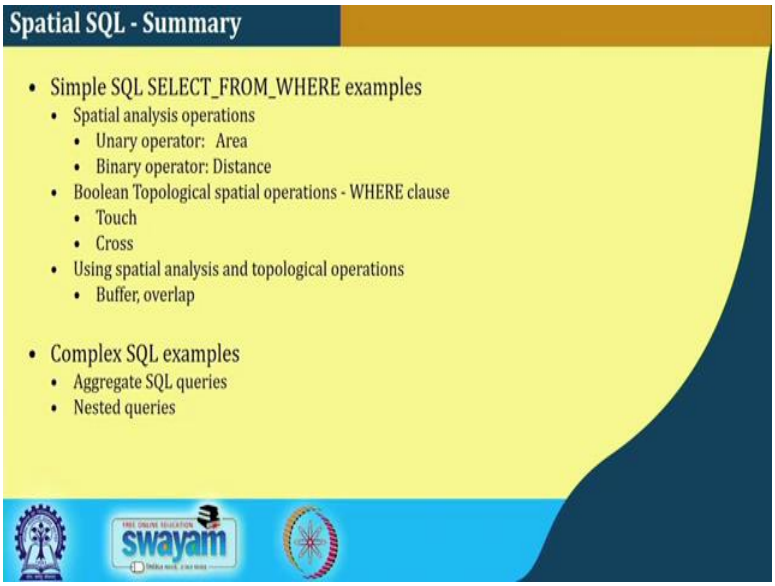


So, we here we beacon SQL 3, so we have user define data types, we can have create type statement, define new data types, attributes and methods are defined, and separate statements for interface and implementation these things are there. So, rather though we require some additional effort at physical data model level right.

So, how do I map those at the physical data model, but there are libraries and the data cartridge and blades where third party library implementing open geospatial standard are available. So, we have lot of other standard almost all user are using these libraries can define their own data types and typing. So, though it seems that you have to define your own data types, but there are different standards which came up right, like OGC standards and different open geospatial standards, so which are being followed by different stake holders and users.

So, in a sense we have a rich state of data types and libraries into the for in place. Rather if you if you recollect the basic data types which are looking for is point poly line and a polygon or surfaces right and composite of these things composite data types right. And where are there are several operations spatial operations which are need to be looked into.

(Refer Slide Time: 03:53)



So, if we try to over all spatial SQL, so for what we have discussed if you try to summarize. So, it is a select from where type of examples spatial analysis operations like unitary operator like areas, binary operators like distances or Boolean topological operations like WHERE clauses where touch cross and this sort of things are required.

And using some of the spatial analysis or topological operations like buffer, over flow and type of things. Like a if you recollect we are trying to see that if on a particular river if it is serving say x kilo meter plus minus from its distance, so I want to create which are the cities it is getting serve by that particular river. So, what we do that a buffer of this particular distance of x we create and overlay with that the city map.

So, what we get that which are the city which are interacting are within that service zone right. So, there are complex SQL example, aggregate SQL queries and nested queries what we have seen in the last class. So, there are we can have aggregate operation, nested queries, we have seen how you can create views and get it get result out of it use it for getting appropriate results. So, these are some of the cases where we have spatial SQL in place.

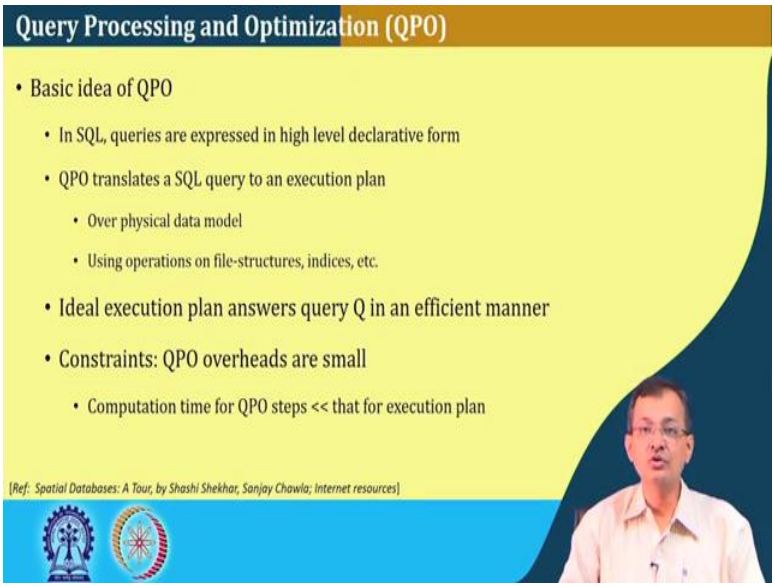
Now, given that as now and then we were discussing that the spatial data when we query it is first of all it is not only spatial type of query there are non spatial or attribute type of queries are there, so the query involve both right. Secondly, this sort of spatial queries

are often both data intensive and compute intensive, so both it is compute intensive and data intensive right which is the typical feature for this type of data set.

So, well processing the query though the result will be same, but the sequence I process the query I execute the query may give different type of efficiency right. So, like some may be processed earlier and type of things right. So, it is important that whether it is possible that while this while processing whether I have a optimize processing phenomena and so.

So, that is why as in case of standard data bases also which is, but in this case more predominant or more it makes more sense to have some keep query processing. And optimization engine or query of processing and optimization approach what we say QPO type of approach things are there right.

(Refer Slide Time: 07:05)



So, that is extremely valid for spatial data rather it is extremely important for this sort of spatial data types right. So, in SQL queries are expressed in high level declarative form right, when we do SQL it is high level declarative form. So, QPO translate a SQL in execution plan right, so that should be execution plan right, over physical data models using operations on file structure indices etcetera.

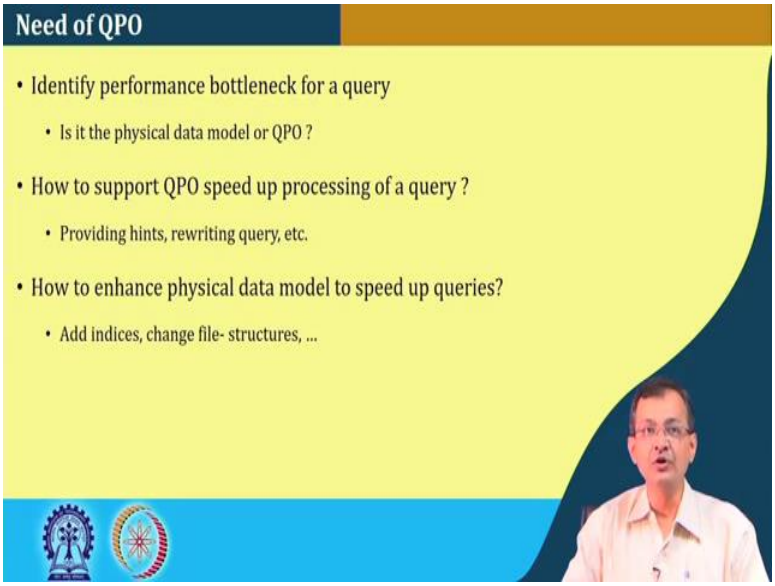
So, this QPO Query Processing and Optimization engine in finding out these execution plan. So, ideal execution plan answers the query Q in a efficient manner or minimum

possible type right. So, if it is the yes suppose given a query I may have two three execution plans right out of that all are correct. In the sense all will give you correct results or expected results whatever you are looking for that query, so all are same.

So, that way there is correctness is not challenging, the problem is that which is efficient in terms of time right, so we may choose that execution plan. So, there is a little but in it, but what do you say that some constant that QPO overheads should be small right, so that should be the constant that.

So, computation time for QPO steps should be much much less than that of the execution plan, in other sense the QPO related processing should not take much chunk of time, so that it effects the overall execution right. So, let us say that is understood right that in order to optimization if the optimization itself in doing the optimizing process doing lot of thing, then it will be a execution plan thing.

(Refer Slide Time: 09:11)

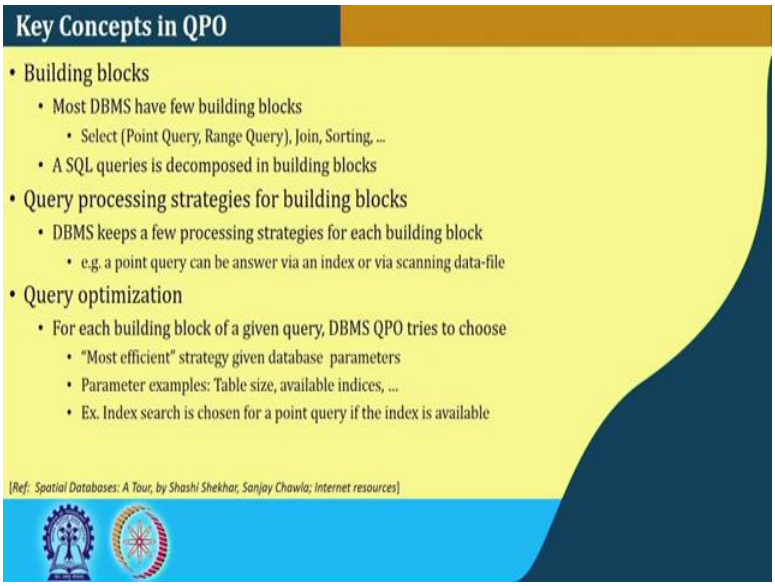


So, need for QPO, identify performance bottleneck of for a query right. So, we need to find out that what is the performance bottleneck of the query, is it physical data models or QPO right. So, whether the bottle neck is the QPO itself or the physical data models, how to support QPO or how to utilize QPO to have a speed up the process of the query providing hints rewriting queries etcetera.

Like as a that is this I need to deal with these are the tables and I say that these tables are more loaded or the data load on this tables are high and some of the tables not high whether that QPO gets that input to have a better execution plan for the for that particular query? How to enhance physical data model to speed up the queries; where there is process there is a way to enhance the physical data models to process the speed up the queries right.

So, add indices change file structures and etcetera right? So these are the things which are needed. These are the we says that there is any need of query processing and optimization things for any data base and most specifically for spatial data base.

(Refer Slide Time: 10:34)



So, what will be the key building blocks? So there are few building blocks in our standard DBMS. That is one is point query select join sorting and type of things right, so these are the some of the building block, so SQL queries is decomposed in to this building block.

So, if I look at the query when I look at the query we will see subsequently the execution tree etcetera that, so these are my building blocks like select, project, joining, sorting and type of things are in building blocks. At the leave level it is connected to the underlining data bases right, if I have the execution tree right.

So, now, these there are these are the building blocks, if I want to optimize I need to optimize with these with these terms of building blocks right. So, query processing strategies for the building blocks is important; suppose it is select, so what will be the query processing strategy? So, DBMS keep few processing strategies for each building blocks, so if it is a select, or join, or spatial join or something some aggregate operations, so it has a some strategies right.

So, like say algo 1 algo 2 algo 3 is for there, so which algo I will pick up based on also dictate that what will be the execution time etcetera. And that is may be based on the type of data sets etcetera right, query optimization for each building block of a given query DBMS QPO tries to choose the most efficient strategy given the database parameters.

So, what are the data base typical data base? On a number of columns, number of cardinalities of the things, over all data base load may be the data base parameters. So, parameter like may be data base size available indices in the data base etcetera etcetera. So, for example, index search is chosen for a point query if the index is available right. So, what we are trying to say that, I for the QPO engine what we have to have the we have to find out what are the building blocks right.

So, and for if building blocks what are the different strategies, or algorithm, or approaches to for that building block. And then optimization may be will be more efficient strategy which will be the more efficient strategy based on the data base parameter. Data base parameter may table size, available indices etcetera, and then we a like a for a index if the index is available. So, find query index search may be chosen as what we say a as the appropriate processing strategy.

(Refer Slide Time: 13:25)

So, there are several challenges in query processing and optimization, so choice of the building block is definitely challenge. So, SQL queries are based on relational algebra this we already know, those who are not so custom the things you may go through any standard book of data base. So, SQL queries are based on a relational algebra, so building blocks are based on relational algebra are select project join.

So, primary building blocks of a select, select project join. SQL adds new building blocks like transitive closure etcetera is SQL 3 that is ok, but these are the traditional thing or three things. So, choice of processing strategies for building blocks constraints too many strategies higher complexity. If there are many strategies for particulars say for the select, or project there are 100 of strategy and then selection of that particular efficient strategy and finding out the efficient strategy itself is a challenge right.

So, commercial DBMS have a total of 10 to 30 strategies 2 to 4 strategies for a every building work. This is a some more much figure there is no a means that is no this what which commercial product I want, but there is a one of thing. So, how to chose the best strategy from the applicable one may use a fixed priority scheme right; say, if the data load is high then I will choose these is the low like this or middle means no significant data load.

Then I choose any stage right that will be some fixed priority, or even I can say that that I can even this type of things like I will execute a project before select, join, before join

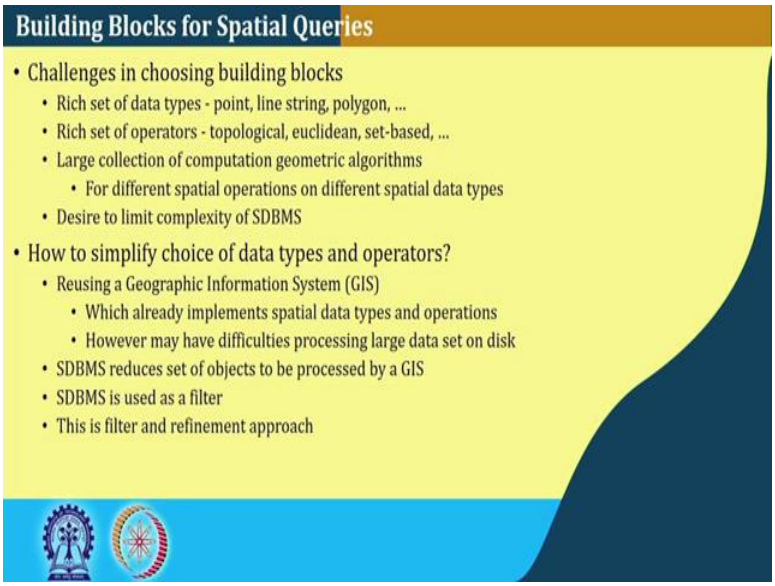
etcetera. So, may use a simple cost model based on DBMS parameter or even looking at the individual building block strategies we can look at the cost model. So, if this is the data base what will be the cost model of the things? If this is the number of, this is the table size, this is the number of column and etcetera etcetera what will be the cost parameter, so that also we can choose from the things.

(Refer Slide Time: 15:40)

QPO challenges in SDBMS building blocks for spatial queries, rich set of spatial data types operations right, a consensus on building blocks is lacking there is a another problem. Like the consensus are that which will be different building blocks and what will be the strategies etcetera is still a challenge; current choice of includes spatial select, spatial join, nearest neighbor type of things, so that is a thing.

Choice of strategies limited choice for some building blocks like nearest neighbor, like we do not have much choice for that. Choosing best strategies, cost model are more complex since spatial queries are both CPU and I O intensive right. So, cost models are so as we talking that both compute and did as in intensive a intensive things are there while traditional queries are mostly I O intensive right. So, in case of spatial query there is both compute and I O and where in this CPU and I O and where this is only I O. Cost models was spatial strategies are not mature right, what would be the cost model for spatial strategies which is are still a challenge.

(Refer Slide Time: 16:56)



Now, building blocks for spatial queries challenges in choosing building block rich sets of data types point line, line string, polygon etcetera etcetera. Rich sets of operators topological operators, Euclidean operators, set based operators there are several of operator. Large collection of computational geometric algorithm for different spatial operation on different spatial data types desire to limit complexity of the SDBMS and there is a spatial data base management system need to be the complexity needs to be reduced for proper or efficient processing.

Now, simplification of the choice of data is an operator, reusing geographic information system may be things which already implements spatial data types and operations. However, may have difficulties in processing large data set of data on disk right, so doing that on the disk is there is there is always a big challenge.

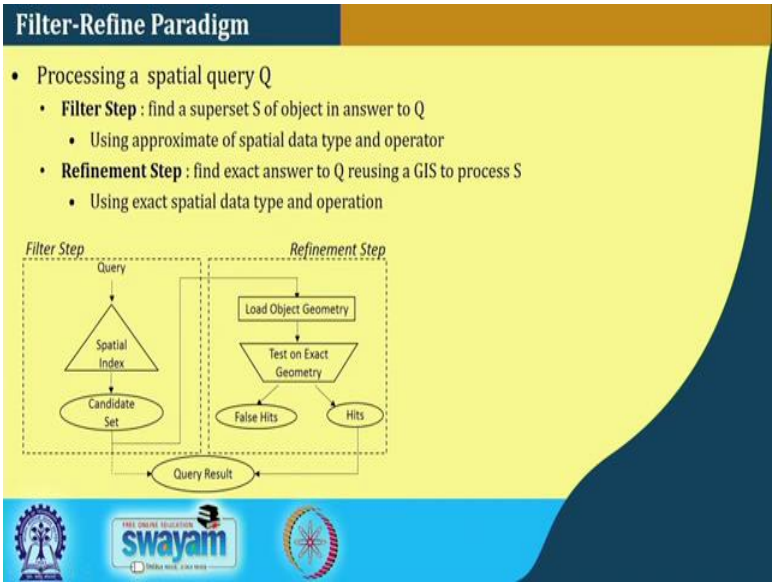
Spatial data base management system reduces set of objects to a processed by GIS. So, the SDBMS is which helps in reducing the set of objects which will need to be process by these GIS, SDBMS is used as a filter the. So, this filter and also, so there is a refine filter and refinement approach, so let us have a summary of that things. So, what we are trying to do? As the building block as there are there are as the your spatial data base are base systems or S SDBMS are both CPU and I O intensive so, we need to have a appropriate strategies on the things right.

Rather for every building blocks we will have strategy we need strategies. But apart from that what we required is that whether I can solve the problem in a say some approach of filter and refine. I bring down the problem in a much smaller scale by filtering and then solve the in a refinement scale to go to that things. So, that filtering is more little gross level, so I do not have all the all the refine stage of the things.

Like I want to see that what is the how much what number of villages are or what now which are the districts with the inundated by the flood in particular region. So, one is that I go on checking that the flood layer with all the districts and doing that, otherwise I can basically check initially and if it is a polygon then it will be problem to handle.

Otherwise we can initially have at the MBR or bounding maximum bounding box level; if they are not intersecting things we will not intersect right. So, I can have a filtering of that which are the candidate things and then we can go for a refinement stage.

(Refer Slide Time: 20:17)

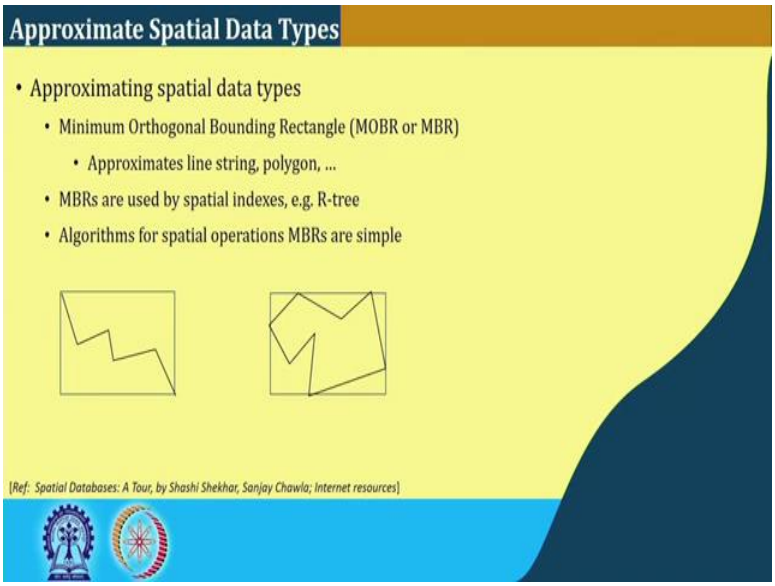


So, that exactly we wanted to looking for, so filter step find the superset S of object in answer to query right. Using approximate of spatial data types and operator and where as in the refinement stage and find the exact answer to the query Q reusing a GIS to process S using exact spatial data types and operations right, so that is the way.

So, if you look at this figure also, so we have this filtering this query is there using the spatial index find the candidates says that which are the probable candidate for the result.

On the other hand we go for refinement step that load object geometry takes exact geometry false hit and there can be actually is the falls hit. And then we can basically put it into the along with that filter step the refinement step basically filter that query results right.

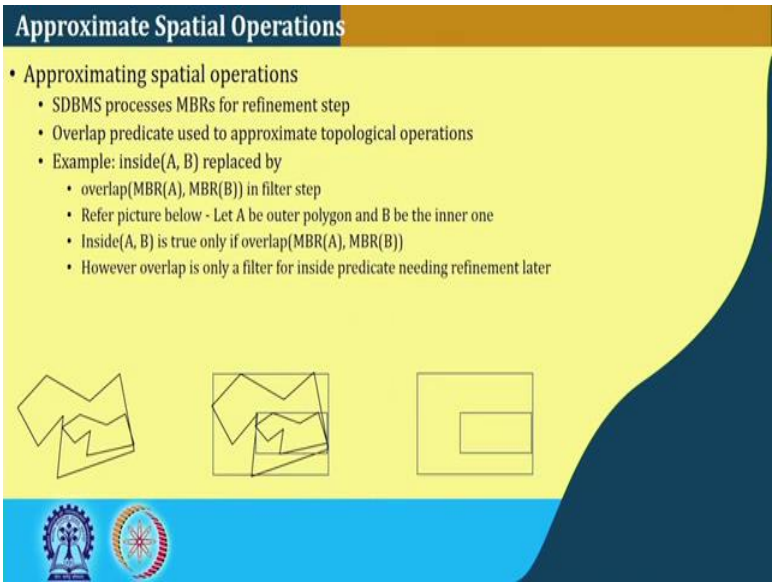
(Refer Slide Time: 21:21)



So, approximate data types spatial data types approximating like we can do the Minimum Orthogonal Bounding Rectangle MOBR or MBR approximate things; that means, the rectangle which we can draw across the boundary. So, if I have a thing I have a spatial line this, so this this the rectangle which is the minimum rectangle which can able to hold that particular object right.

And similarly for the polygon things like that right, so this is not a very difficult to find out that what will be the rectangle, because only appropriately sorting that the co ordinates of that age is. But he says I can find out that is the what will be the step of the rectangle right. But looking at the individual object and comparing them whether they will upon not say challenge it is a very time consuming thing.

(Refer Slide Time: 22:28)

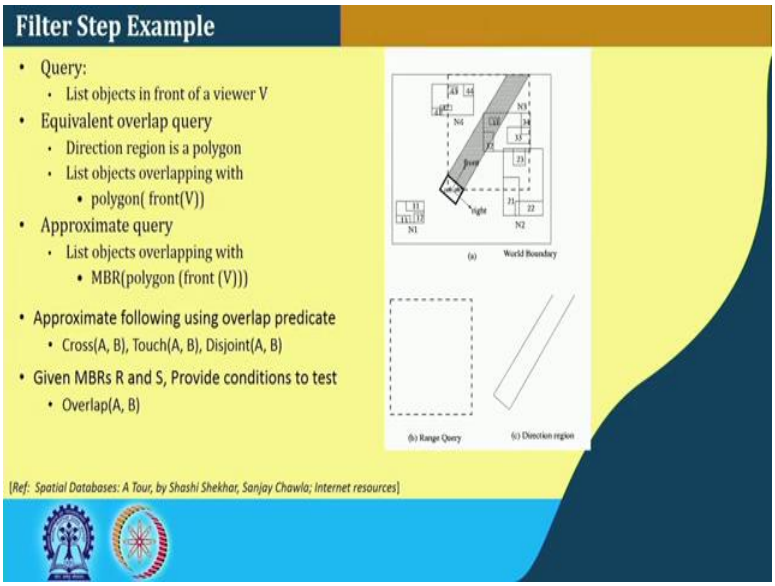


Now, if they SDMS as processes the MBRs for refinement step overlap predicate used to approximate the logical topological operations. So, example inside A B replace by the MBR of A and MBR of B in filter step. So, let A be the outer polygon and B be the inner polygon, then inside A B is true if the if they overlap right. However, overlap is only a filter for inside predicate needed for the refinement later.

Now, let us see there what we are trying to look at. So instead suppose I have irregular two polygons right, so I want to find whether this is inside that or not. So, what we do? One is that I can go point by point full inside or not an kind to check, otherwise I generate both the MBR for this, suppose this is the MBR for the outer one this is the MBR for the inner one right.

Now is if this MBR this MBR is fully inside or embedded within that larger MBR then I can say the object inside the thing is also embedded. Now, working with rectangle is much much easier than doing any computation with the irregular say polygon right. So, in the filter stage I do it like that, but in the refinement stage I basically what we want to do? We basically now look at the in the in the actual polygons in this case. But you see my search place are reduced to these thing only, so I do not have to search the whole data sets and type of things, so it is much easier and convenient to work on.

(Refer Slide Time: 24:26)



So, this is a example in earlier we will find in the book, so list the objects in front of a viewer V. So, V is the viewer, so equivalent overlap query direction region is a polygon, so I want to find out that what is is the front of things, so this is a polygon right. List objects overlapping with polygon front of V, so I define front of V is a polygon and then polygon this is a polygon and then I want to find out what are the object is overlapping. So, approximate query list object overlapping with thing is the MBR of polygon front ofV.

So front of V find out the polygon approximate following the overlap predicate cross A B, touch A B, disjoint A B right, so these are this doing on that particular MBR itself. And then the given MBRs R and S provide conditional conditions to test like overlap A B or not right. So, see what we can do, we refine it with respect to the MBR which is much easier, it is a rectangle. And then refine it based on that how much what is the actual touching that particular polygon or not.

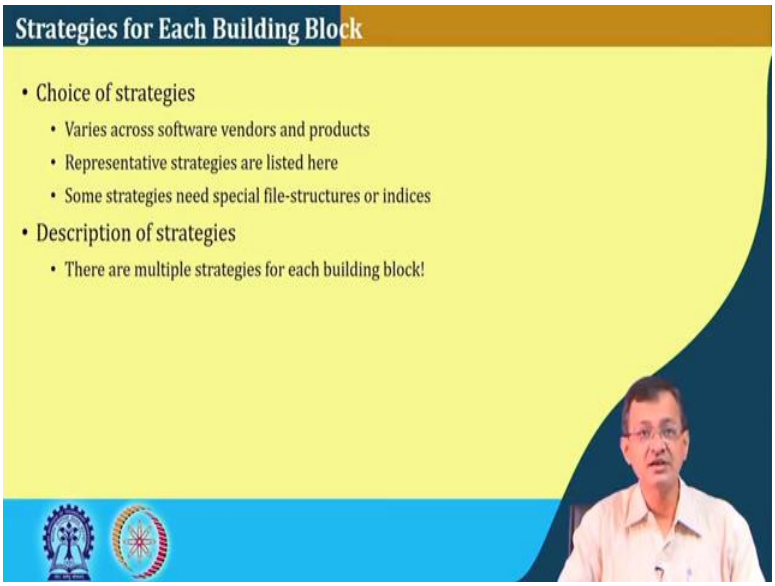
(Refer Slide Time: 26:00)

Now, choice of building blocks is a definitely a major challenge where is across software vendors and products represent there are few representative building blocks, but mostly varies across the software. List of building blocks one for point query, name a highlighted city on on a digital map return one spatial object out of a table.

Range query list all countries crossed by the river particular river say Amazon also and return several objects and return several objects within a spatial. So, what we are having? We having point query which has a particular highlighted objects in the map, we have a range query that I want to find out all the in a particular range that all the countries cause by a particular river example.

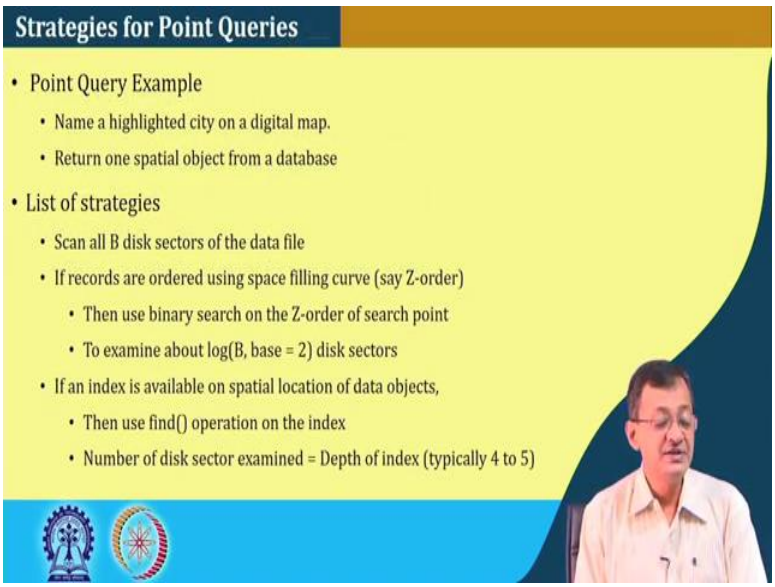
So, it returns the objects within the spatial range of that particular table spatial join it is a very tricky thing. List all pairs of overlapping rivers and countries returns pairs, pairs from two tables satisfying a spatial predicate right. So, it is a like we have a normal join in case of a spatial join two spatial table contain spatial data need to be join. So, list all pairs overlapping a rivers and countries. Nearest neighbor find city closest to the say particular say in this case it is to be Mount Everest, so city closest to a Mount Everest. So, what we will do? Return one object from the collection, so I have to say this city is the nearest to the Mount Everest.

(Refer Slide Time: 27:38)



So, strategies for each building block, choice of strategies varies across software and vendors, vendors and products. Representative strategies are listed we will see some of the representative some strategies need special file structure or indices. So, there are multiple strategies for each building block there can be in multiple strategies for each building blocks.

(Refer Slide Time: 28:01)



So, like we say some of the strategies like query things like name a highlighted city on a digital map return one spatial object from a database right. So, scan all B disk sector of

the data file, so I scan all the disk sector. If the record is ordered in a space filling curve in a Z order curve we will see that what does it a X order curve or the space filling curve means, so is the space filling curve.

Then use the binary search on the Z order curve right or if there is some way it is the thing is arranged, then we will use that. To examine log B best two type of this sectors to find out that where the things are there. If an index is available on the spatial location of data objects, then find operations on the index. Number of disk examined is the depth of index typically 4 to 5 should be the typical usually the depth of the index.

(Refer Slide Time: 29:01)

So, similarly we can have a range query things right, list all countries crossed by the river some river Brahmaputra return several objects within a spatial region of the table. Here also we scan all those B sector from the things if the records are ordered in space filling curve then determine the range of the Z order.

Because now are the range of value satisfying that particular range query use binary search to get lowest Z order within the query and scan forward the data till the highest z order, so I want the only thing. Even index is available on the spatial location of the object, then use the range query for the operations right. So, that is so we can do for the range queries.

(Refer Slide Time: 29:50)

Spatial join is a little tricky or not that is the interesting also; list all pairs of overlapping rivers and countries right return pairs of two tables satisfying a spatial predicate right. So, list of strategies that can be a nested loop, I check tested all possible pairs for spatial predicate, all rivers pairs with the countries and type of things or space partitioning a test pairs of objects from the common spatial region only, rivers in India are tested with in the India only, so there is no question of taking the rivers of say Australia testing with India.

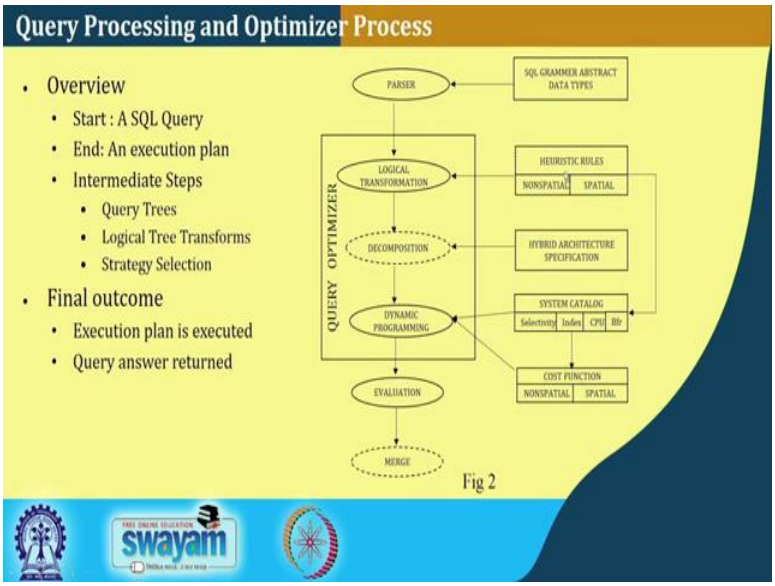
Tree matching I can have hierarchical pairing of object groups in each table right; other there is spatial join index, external plane sweep types of things there are different other strategies we can for. So, there are different type list of strategies which we can follow, now this strategies out of that one strategy to be followed for this spatial join right.

(Refer Slide Time: 30:49)

Similarly, other category is that nearest neighbor, so find the city closest to say a Mount Everest as well as see return one spatial object from city data. So, there can be again list of strategies two phase, like Fetch Cs disk containing the location of Mount Everest. Minimum distance, Mount Everest cities to the fetch sector; test all cities within the distance of the M of Mount Everest range query right.

Second phase approach recursive algorithm for R tree; eliminate candidates dominated by some other candidate's right. So, we can eliminate candidate dominated by other candidate, so there is a single phase approach as well.

(Refer Slide Time: 31:33)



Now, when considering all those thing I can we can have a query processing and optimization process like I see here start with a SQL query which is as a particular grammar and data types. And it parser goals for a logical transformation using heuristic both non spatial and spatial, and then decompose the query for the hierarchical specification.

And we can use system log for selectivity index CPU or buffering of the system log and we can have custom cost functions like what will be the cost of spatial function, what will be the cost of non spatial function on the and do some dynamic programming, then evaluation of the things and merge the data if at all if the queries are there. So, if you say start with a spatial query end with the execution plan, so I have a execution plan of the things intermediate step, query trees, logical tree, transform strategy selection and so on and so forth and final outcome is a execution plan to be executed query answered returned.

(Refer Slide Time: 32:42)

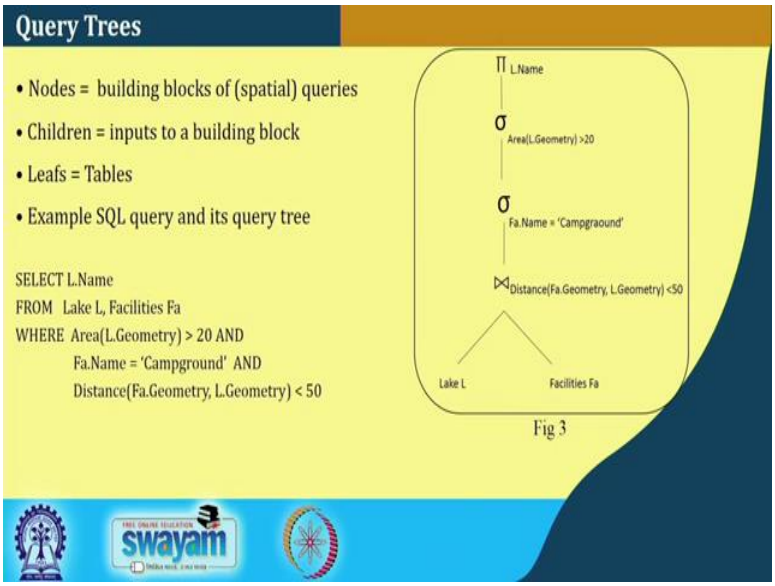


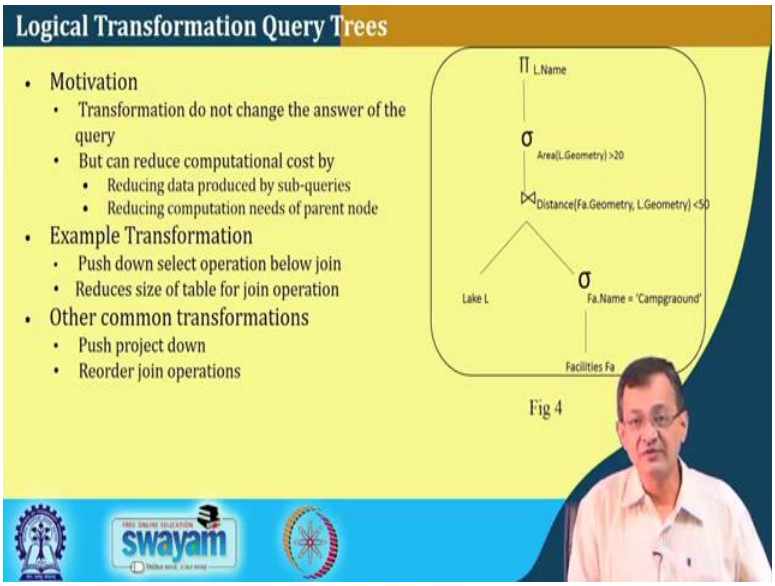
Fig 3

Like we just see one example, like we have that like if you refer to the other previous things. So, we have nodes, building blocks, children input of the input building blocks tables and so on and so forth. If might query select L names from lake L and facilities Fa and area La greater than 20 right. Some 20 units and the facility name is campground, and the distance between the campground and the lake is less than 50 some unit.

Now, if it is this is my query this is one of the execution see, so the at the bottom the lake L and facility Fa are there which connects to the data base. I want to find out from the distance right within this 50 things, I want to find out the campground which are the campground and then the area calculation of the geometry right.

Now, you see this is my execution plan which gives and finally, projecting that L name and type of the things right. Now, I can have different many spatial things I can do some of the things earlier some of the things thing. And the thing based on these different costume of this particular strategies I can have different type of over all computational cost right. Though by results will be same, but based on the different type of execution tree I can have different type of results.

(Refer Slide Time: 34:06)



We will come we will discuss those things in my same thing I can have a different execution tree. So, what we what we tried to look in today's discuss any is that first of all the spatial data needs spatial consideration because of this different type of operations large data types both computational and data intensive. And also is there is need for a query processing in optimizations right approach for this things right. We will continue our discussion on this particular aspects in our subsequent lecture.

Thank you.