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

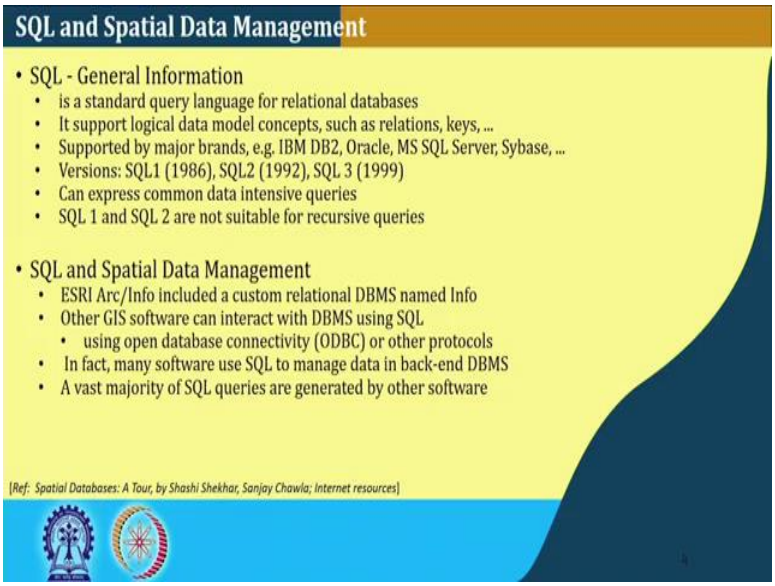
Lecture - 13
Spatial Query Processing / SQL (2)

Hello. So, we were discussing about several aspects of spatial informatics, rather if you recollect that last class we started with we are discussing about spatial query processing or how we can processes queries into the things. Like any information system, the query processing plays a important role right. So, as we have discussed it is not only retrieving, I can input data and manipulate data and type of things. So, end of the day for the end user or even that the other in system, this SQL plays the important role right. So, what I believe that you have some fairly good knowledge on normal databases, relational databases and SQL. So, we will build up on that right.

So, the obvious query will come that why if there is; there is already a database system and SQL is in place, why you require a spatial database a spatial SQL for that matter right. So, why you are why we require spatial consideration for SQL for spatial data, that we will try to see over looking at different type of examples, different type of approaches and type of things right. So, we should again keep in mind this spatial databases are pretty large in nature and they have both computationally intensive and data intensive right.

In number of cases I may not be able to pull the whole data in the main memory to query process the queries right I need to have some other mechanism to do that. And with this context let us see few example scenarios last class also we have seen some of the things we will be primarily referring to very that standard book of professor Shashi Shekhar and professor Shashi Shekhar for this purpose also right.

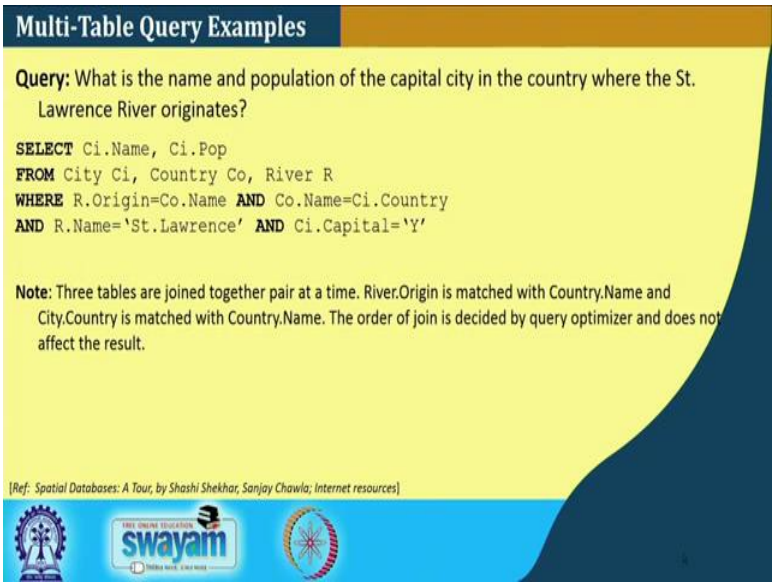
(Refer Slide Time: 02:27)



So, SQL. So, if you have a general very generally it is a standard query processing for any relational database is known to us it supports logical data concepts like relationals, primary key, foreign key and all those things supported by a major bunch of vendors from Oracle to IBM to sequel server to Sybase to MySQL to open any open source SQL any open source database engines. So, it is a de facto language for any database there are several versions which are there right SQL 1 2 3 can express common data intensive queries right it can SQL 1 2 are not suitable for recursive queries as we know SQL 3 we can have.

And if we now like to see SQL and database management system. So, there are several means GIS measure like ESRI and others which include customize relational database named called info which other GIS software can interact with DBMS with SQL and many software uses SQL to manage data in the backend of the data DBMS right. A vast majority of SQL queries are generated by other software right. So, it is basically I have a upper level layer applications and then I am basically firing those queries to generate something at the lower level at the backend to look at the databases.

(Refer Slide Time: 04:02)



So, like we have seen some of the queries few more queries like name and population of the cities which are where the some Saint Lawrence river originates or say a river Godavari originates or something in our context. So, I say SELECT Ci NAME, Ci Pop FROM city Ci and Country Co and River R where R dot origin equal to Ci dot Name Co dot Name equal to Ci dot Country and Name of the river and capital equal to Y. So, if the query where population of the capital city of the country, where this particular river originates right.

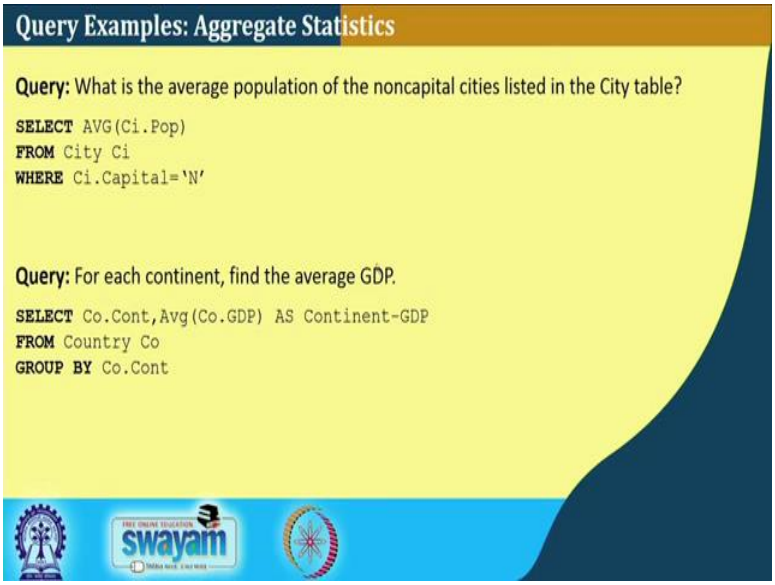
So, I want to find that particular City which is a Ci dot Capital equal to that characteristics of the Capital is true and find out that the river origin is that country origin, Country Name and country name also belongs to that particular city and then and your particular river name is there and so, and the capital city is equal to Y. See this particular thing is something which joins multiple table and it is basically not using any spatial property per se right.

I can have databases right a say standard database tables and I can join this join this tables. So, it is basically looking at the query which is looking at the multiple queries. So, in this case three tables are joined together at a time River dot Origin matched with the Country dot Name and the Country name is matched with the Country dot Name, and the a and the order of the join is decided on the Co query optimizer does not affect the

result whether I do this one first or this one first on which sequence, it does not it does not affect the result right.

So, but as such I am not using per se any spatial property like I am not using any geometry property or any other type of things or queries which have multiple tables to join.

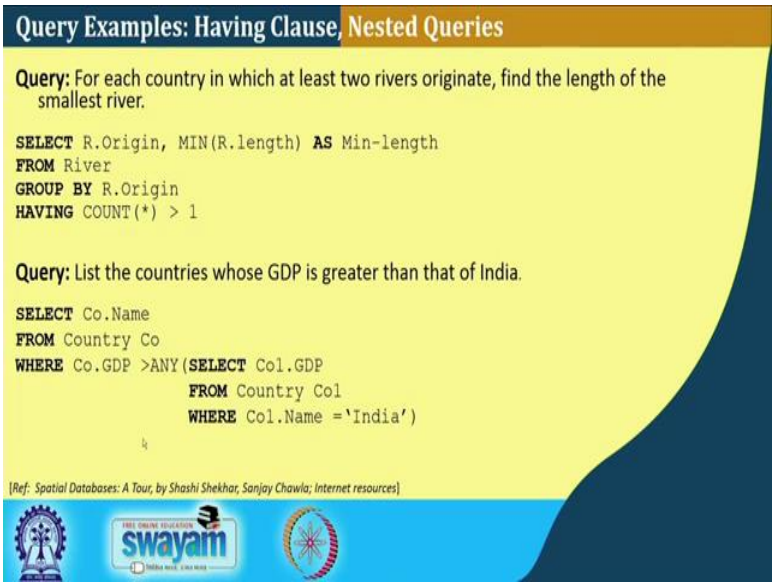
(Refer Slide Time: 06:18)



And we can have something aggregate statistics queries right where the average population of a noncapital listed in the city table or something like that right what is the average population of the noncapital cities listed in the city table. So, I say that SELECT average Ci dot Population if I want to print the city name also, then Ci dot name coma city average Ci dot population from city Ci and Ci dot capital property is known. So, all noncapital cities right.

Now, we can say for each continent find the average GDP. Similarly I can have say average GDP of the things and that Co dot Continent where country belongs to the particular continent and then aggregate that and as map it as a continent GDP, where and group by the Country dot Continent. So, what we are doing is more of a aggregate operations right for every continent we are finding those or those embedding countries and aggregated their GDPs and finding the average out of it.

(Refer Slide Time: 07:29)

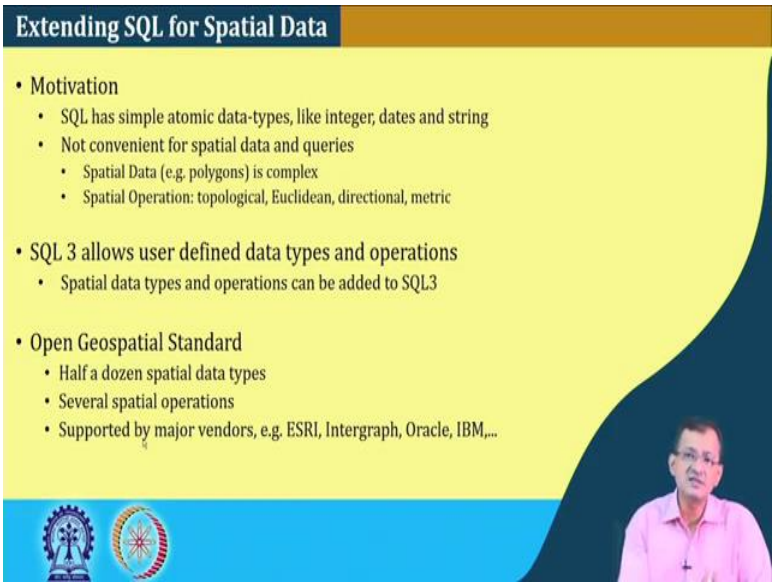


So, there are examples like having clause nested queries like this. So, like for each country in which at least two rivers originate find the length of the smallest river right. So, this is little complicated queries, but nevertheless it is again solvable by our standard SQL knowledge right. So, like origin R dot Origin minimum r length AS min length and from river group by R dot Origin and having country star greater than 1. So, what it is telling that at least count star greater than 1. So, at least two river origins so; that means, it should be more than 1 river.

Similarly, list the countries whose GDP is greater than that of say India right. So, select say particular Country dot Name from the country table, where Country dot GDP is greater than any select any country CoI dot GDP and type of things right so; that means, this is the syntax what I am trying to see the list all those countries whose GDP level is greater than that of India. So, that can be India or any other country for that matter right.

So, again what we want to insist that here we can have like we have seen nested copy query, multiple table query a this things are having clause and type of things, but never the less we are; we are able to do with our standard database knowledge or standard database table format or so, to say non spatial database format. We are not exploiting any data out of it any spatial property out of it, but nevertheless we can able to have those queries in place.

(Refer Slide Time: 09:16)



Now, extending this SQL to spatial database. So, far see these examples I am not exploited any spatial property right or neither any spatial property is utilized here. So, what we are trying to did, what we did that the tables are having as good as our normal tables of a library or bank or something and we can use same type of SQL things to capture this table right.

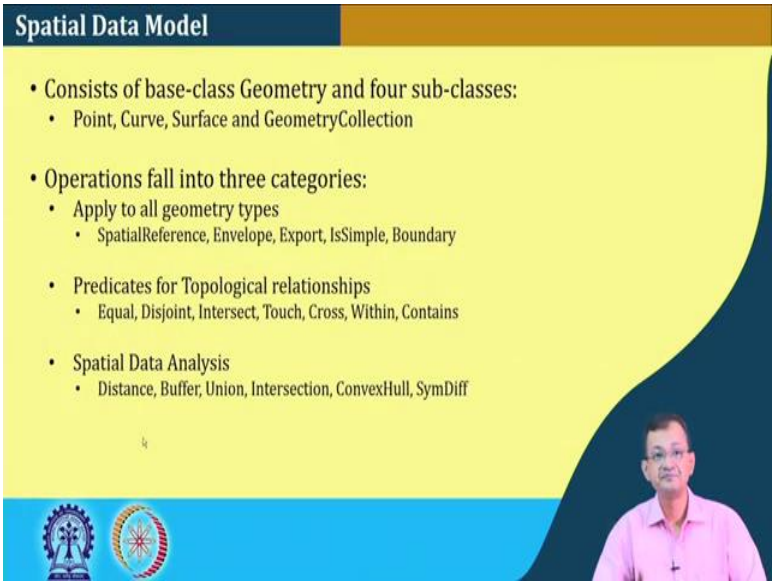
So, extending SQL for spatial data. So, SQL can simple atomic data types like integer, dates string these are the simple or standard datatypes not convenient for spatial data queries, where the spatial properties or spatial concepts are involved right like a spatial data type polygon is a complex stuff, spatial operation like there are operation like what we have seen like topological, Euclidean, directional metric based operation these are different spatial properties which are in this place.

So, SQL three allows user defined data types and operations. So, what I require some user defined data types need to be there I need to define what is a polygon, I need to define what is a polyline or point and this different type of topological relationship. So, spatial data types and operations can be added to SQL 3. So, open geospatial standard half a dozen spatial data types right several spatial operations supported by major vendors of this who are working in this spatial area.

So, open Geo geospatial standard or which are supported by OGC open Geospatial Consortium and it has a different type of spatial data types which are supported several

spatial operations and also supported by a major players or the major vendors in the areas of spatial database or spatial informatics.

(Refer Slide Time: 11:23)



So, addition to whatever the things is there in the SQL, it has a geometry class what we say base class and there are typically four subclasses like one is point, one is polyline or curve, one is surface or polygon and geometry collection. So, again we are telling that we are now we are primarily into the looking into the your 2D type of surface right or two dimensional space we are talking about. So, point polyline or curve what we say and the surface or what we say polygon which are is there along with there is many geometry collection, this combination of one or more of this basic geometry types.

So, again if we look at the operations, all the operations apply to four geometry types. So, spatial operation there are three major categories right one is spatial reference. So, one is there are major three categories like app application of the geometry types, another is topological relationships and the data analysis. So, what we see that what whatever operations we want to do with the spatial data, we can put this into this three categories either it is working with different geometry types or something which is having a topological relationship.

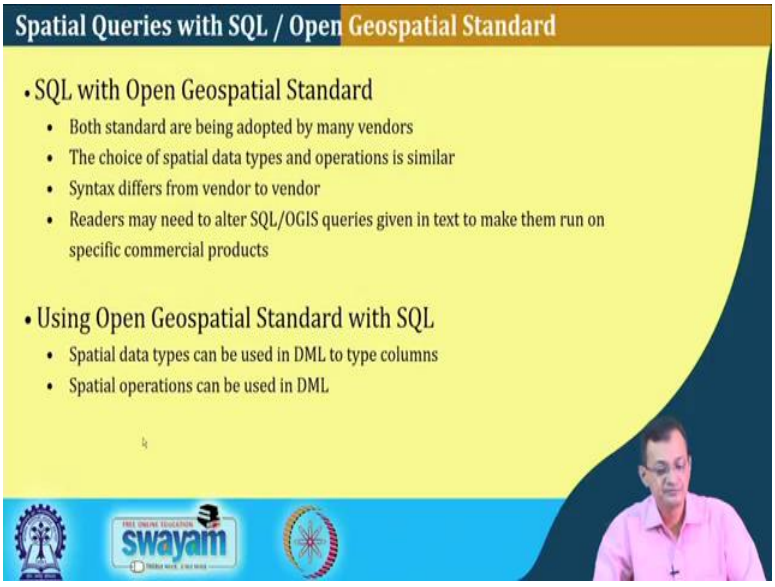
As if you recollect we discussed about the topological relationship like touching, intersection, in inside the things or a different type of cases what we have seen or crossing and or not intersecting and type of things right and there are; there are data

analysis type right. So, if you look at the geometry types those there is a requirement of the spatial reference system, envelope or the boundary box export type of things where you can debt.

So, there are simple data types or there are boundary type of consideration and when we have a topological relationship as we have as we have seen equal, disjoint, intersect, touch, cross, within, contains these are the different topological relationship. If you look at the data analysis type of thing distance, buffer, union, intersection, Convex Hull, SymDiff and different type of things which are more of a some of the concepts coming from the computational geometry point of view. So, which helps us in spatial data analysis.

So, if we try to look at this spatial operations in some part or other this will fall into. Now when we talk about the spatial queries or spatial query engine or a spatial SQL, they should able to serve this type of things right one is that data types and the corresponding operations on those data types right. So, my SQL other in addition to that whatever we can do with the SQL should able to support this.

(Refer Slide Time: 14:41)

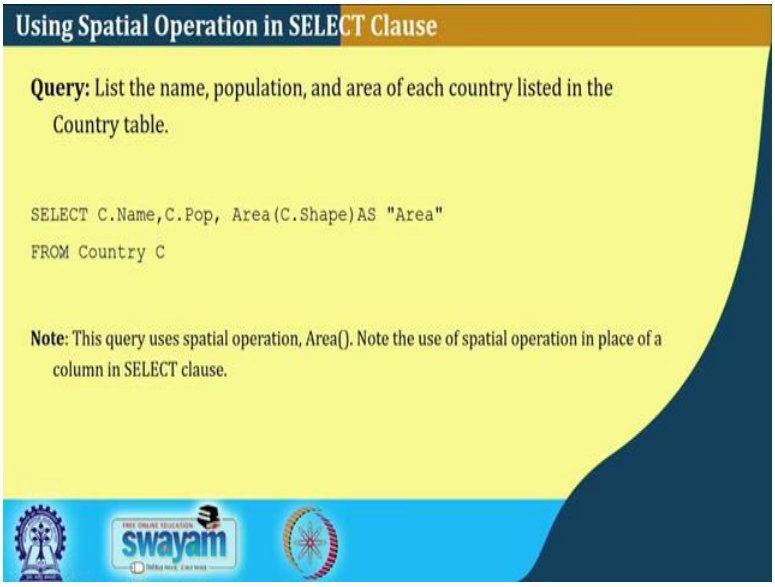


So, spatial queries with SQL and open geospatial standard. So, SQL with open geospatial standard. So, both standard are being adopted by many so called major companies or vendors. The choice of the spatial data types and operations is similar, syntax differs from one type of package to another, but the conceptually things remain same. So, we

should have a may need to or the programmer need may need to alter the SQL OGIS given in this whatever we will be discussing here on to specific products and using open geospatial standard to SQL the spatial data types can be used to in DML to type columns or in spatial operations can be used in the data manipulation language etcetera.

What we mean to say that you can use this SQL for writing or retrieving data or for data manipulation and different type of operation in other sense this should be as versatile as the as our non spatial SQL.

(Refer Slide Time: 15:56)

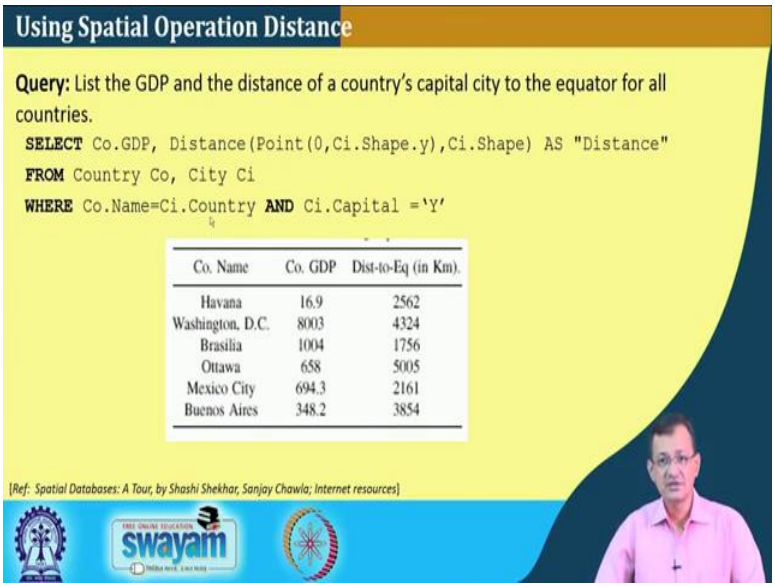


Like here, now we look at some of the queries which are have some spatial context or some spatial concepts which will be there like list name population area of each country listed in the country table.

So, it is SELECT C dot Name, C dot Pop, Area C dot Shape AS Area Country C. Now this area is need to be calculated based on the polygon of that country or the encompassing boundary of the country. So, the area calculation needs to be done based on this based on the spatial context like in this case a polygon finding the area of a polygon. So, that says operation which is under the context of the spatial operations right. So, the query uses spatial operation called area note that, use of spatial operation in place of a column select clause right see we are using this spatial operations here itself area C dot Shape right. So, this operation is being in that SELECT clause.

So, select if you remember we select from the columns right. So, I select that C dot Name C dot Pop and then area on C dot Shape C dot Shape is a column and I am calculating the area based on the shape of the things right.

(Refer Slide Time: 17:34)



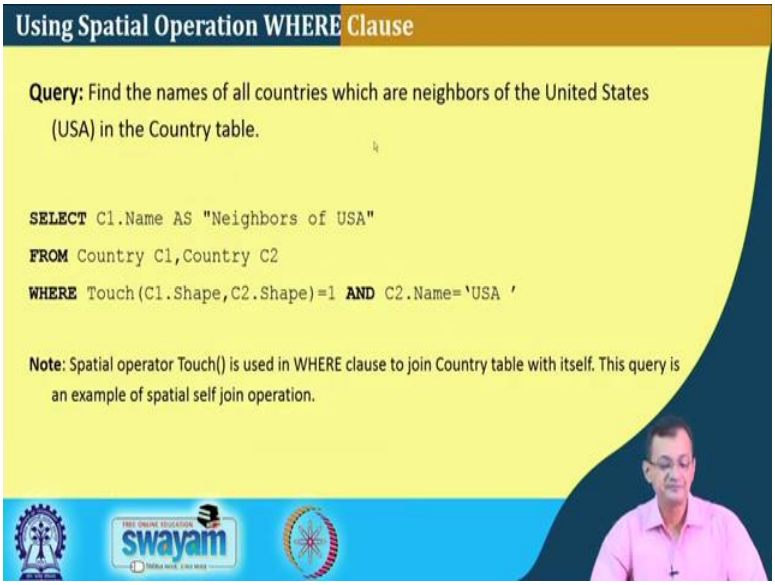
So, this is one spatial operations. So, here list the GDP and distance of the country's capital city to the equator of all countries right. So, something which one the DGP and the capital cities distance from the equator right.

So, this is the way the query means may not make sense of the applicability, but if this is the query right. So, two things are there GDP that is a non-spatial stuff, countries capital also I have we have used that now what the making the distance from the equator required some geometry operation or spatial operations right. So, Co dot GDP distance point 0 to Ci shape dot Y right. If you see that x if I my xy is the latitude longitude. So, if I said that 0 and that the Y value is give me the distance of the things right and as distance and country Co and city Ci. So, what we require? Country name equal to Ci dot country right and cities country is the same and city dot Ci dot capital equal to yes; that means, the city should be capital and it belongs to that particular country right.

So, in this case what we use is a see first of all finding out that point and finding out the distance from the particular Ci shape right. So, that is; that is important to see. So, this one is a spatial operations right. So, it has to be on the spatial operation. Now at least that particular country, the GDP and distance from the equator right. So, see we are this type

of queries if you recollect it is not possible our in our standard SQL with the standard database and type of things right. So, this is feasible with our this sort of stuff.

(Refer Slide Time: 19:48)

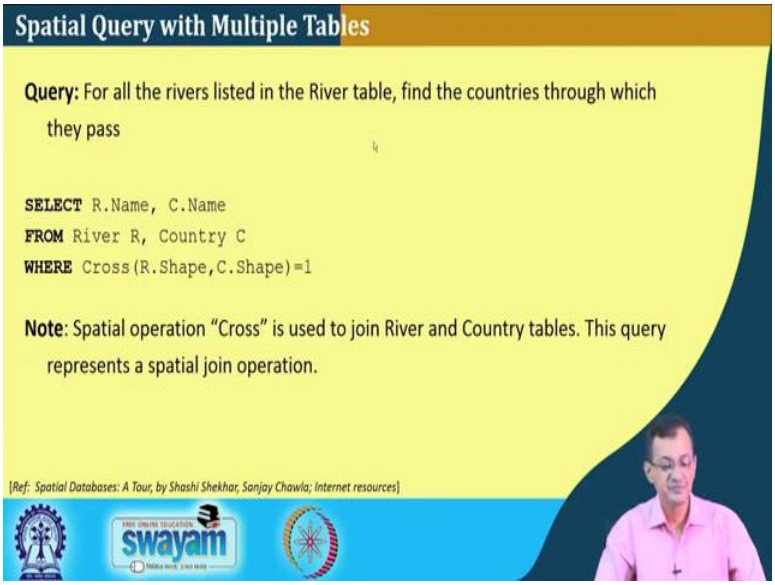


So, we can have spatial operation with where clause find the names of the of all countries where the neighbors of the who are the neighbors of the United State in the country table or neighbors of India on the country table. Now there should be a definition what whom you called your neighbor right. So, for our case if a country touches another country or the polygon touches one polygon then they are neighbor. So, spatial operator touch is used where the clause to join country table with itself. These query is an example of spatial self join operation right. So, it is a example of a spatial self join operation right.

So, what we have C dot name as neighbor of USA or even any country. Country dot Ci equal to country dot C2 then touch C1 shape coma C2 shape. So, it is a self joining on the country table country table and C2 name is equal USA; that means, one of the thing is a USA right. So, here what we try to look at see here the touch is a topological function. So, we are exploiting touch where of the self joining if those are touching one of the countries USA. So, then we say its a we have a they are the neighbor that is my definition of neighbor, I defined neighbor as that one will touch another thing right it can be other definition of neighbor like if I say not only touch within this distance also is neighbor and type of thing so, that becomes a thing.

What is what we want to look at it that we require this something at topological function which is something different from our standard way of looking at it.

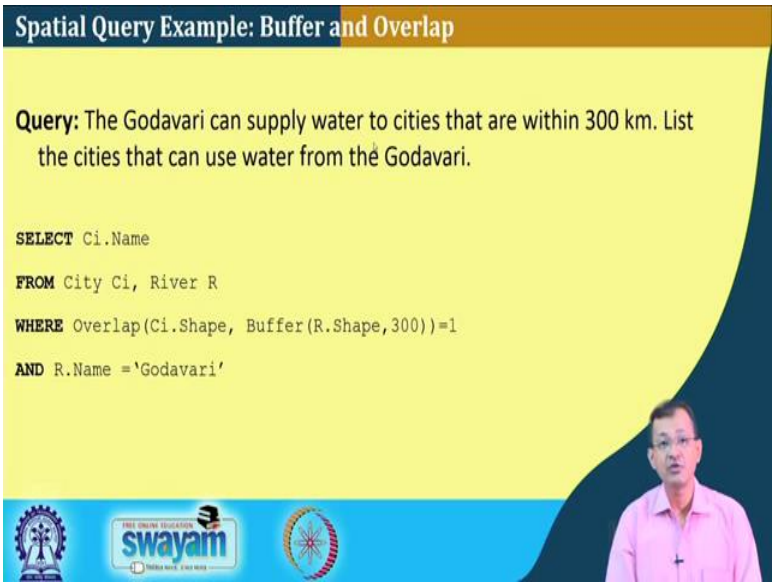
(Refer Slide Time: 21:36)



Spatial query with multiple tables right. If I for all rivers listed in river table find the countries through which they pass alright. So, what we want to look at that for all the rivers which are listed in the river table, find the countries through which they pass right. So, select R dot name C dot name River R and Country C and where cross R dot Shape coma C dot Shape equal to 1. So, very straight forward. So, we have the river name and the country name from the river table and the country table and my this joining criteria is that river and the particular country should river should cross; that means, river dot shape.

So, river if you consider polyline then its a polylines here or if you consider a again as a polygon or something. So, it is the that shape it will take up and country dot shape they should join or they are crossing should have a two value or the one value. So, spatial operation cross is used to join river and country tables this query represents a spatial join operation. So, this a typical thing is a spatial join operation. So, we have a spatial join.

(Refer Slide Time: 23:06)



Spatial query example say buffer and overflow, the Godavari can supply water to the cities which are 300 kilometer within 300 kilometer within it right.

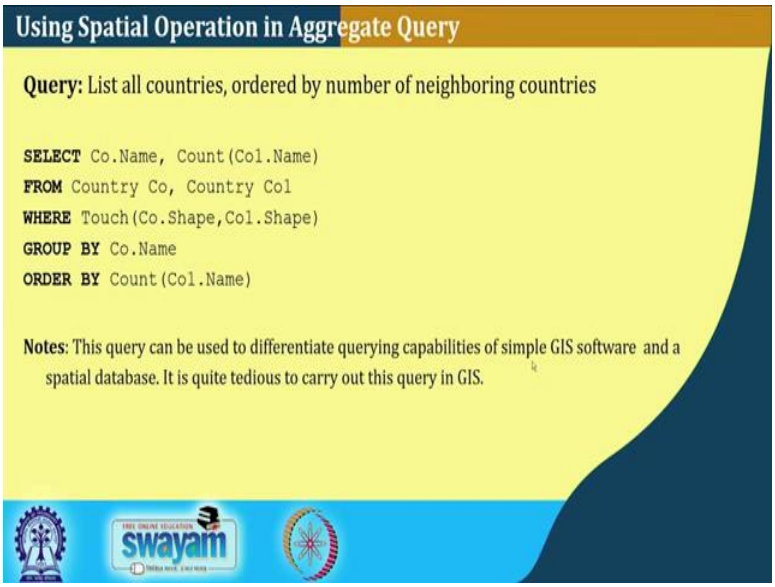
List the cities which can use water of Godavari say this is my query. So, I have a river Godavari which can supply water to cities which are within the 300 kilometer. So, whenever the cannel or river is there. So, we find that what is its means serving area or sometimes I am not very good at hydrology or that type of stuff. So, I can say that what is this catchment area how much it is can serve or what is the thing. So, we define that it can go upto 300 kilometer anyway that is the thing. If it is there then we want to list those cities which can use the water of Godavari right.

So, if C1 dot name city Ci river R. So, it is overlap C1 shape buffer R dot shape 300. So, this is a another challenging equal to one and the river name equal to Godavari. So, you see what we are doing. So, what is our objective? I have a river and I have different cities across around the things. So, from the river I want to find out within the 300 kilometer which are the cities lying. So, what I what I do? I have this river and along with that I create a buffer for that river right. So, 300 kilometer buffer now this buffer is a polygon.

Now, this polygon I see that if the city shapes are all points or even polygon does not matter, if it is a point then I do a overlap and if they overlap if the overlap is 2 or 1 then I say that particular city is can be served by the Godavari river or can be can receive water from the Godavari river particular thing right. So, what we see that overlap C1 dot shape

and buffer of this river R within the 300 kilometer and into this. So, this is again a spatial operations say again we can understand that this is not the feasible or this is be they are in case of a non spatial or our standard SQL type of operations.

(Refer Slide Time: 25:40)

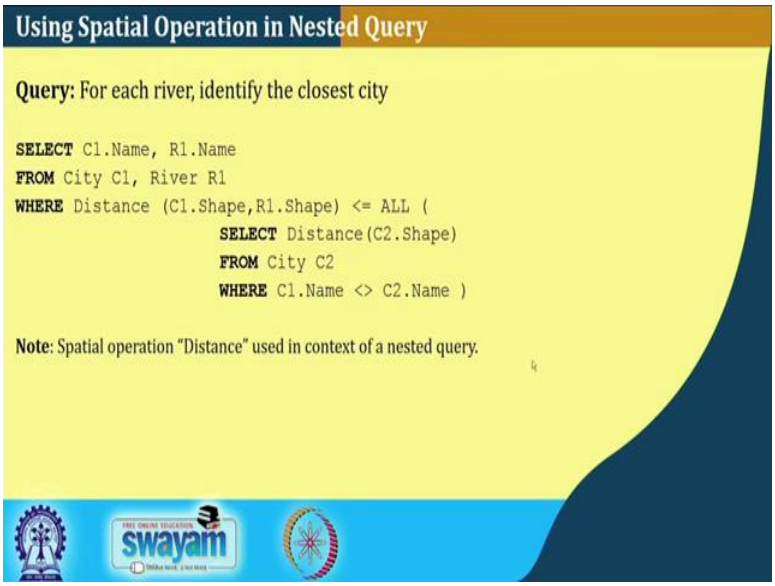


Then using spatial operation for aggregate query. Find all countries ordered by the number of neighboring countries right. If I want to order by the neighboring countries the number of neighbors you are having. So, it may be ascending or descending and whatever.

So, this is little bit complex. So, it is aggregating. So, I need to find how many neighbors are there and type of things. So, what we see Co dot Name and Co count Co 1 that is another instance of the country of the name equal to country Co country Co 1; that means, something self what we are doing self join then first of all in order to make them neighbor. So, touch Co dot Shape coma CoI dot Shape should be touching each other and it is grouped by and that Co dot name and ordered by the count of Co 1 dot name right.

So, I do a self-join of the country and then see that whether they are touching each other their neighbors and then try to look at their count their neighbors and try to sort them or order them according to the neighbor right. So, by default it may be ascending if I want the descending then I have to specify that particular key or decend or DESC where we have the order by clause right.

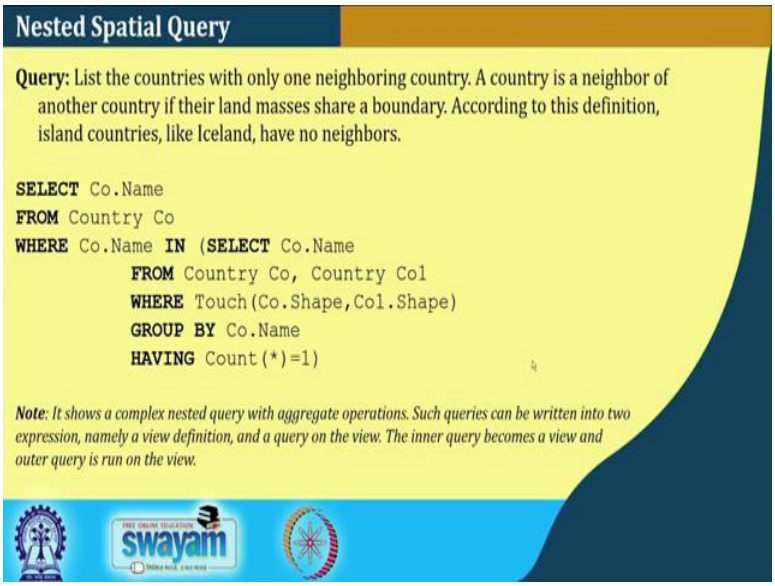
(Refer Slide Time: 27:08)



So, there is another query operations in nested query right. So, select C1 dot Name R1 dot Name city C1, River R1 and then. So, what we are doing for each river identify the closest city. So, little more complex right. So, for each river what is the closest city right what is the nearest city. So, one is the distance what we have do? To either from the I have to have look at the city table and the river table, then distance between C1 dot Shape and river less than equal to all and the select distance C2 shape from city C2 where C1 name is not equal to C2 name right.

So, what we are trying to look at that taking another instance of the city table and try to find out that what is the distance between this river and the city. Now this distance function calculate that a spatial for calculating the distance from the river right. So, spatial operation distance used in this context of a nested query.

(Refer Slide Time: 28:20)



Nested spatial query list the countries. So, another example list the country with only one neighboring country, a country is a neighbor of the country if its land masses share the boundary or the land masses touches right. So, in other sense like I can say that Bangladesh is a neighbor of India, but Srilanka in that sense is not a neighbor of India according to this definition because they are land masses are not touching one another right or something else in some other part is not a neighbor of the country right. So, if we even look at it that way. Never the less if the land masses touches my neighbor definition is they are in the same right.

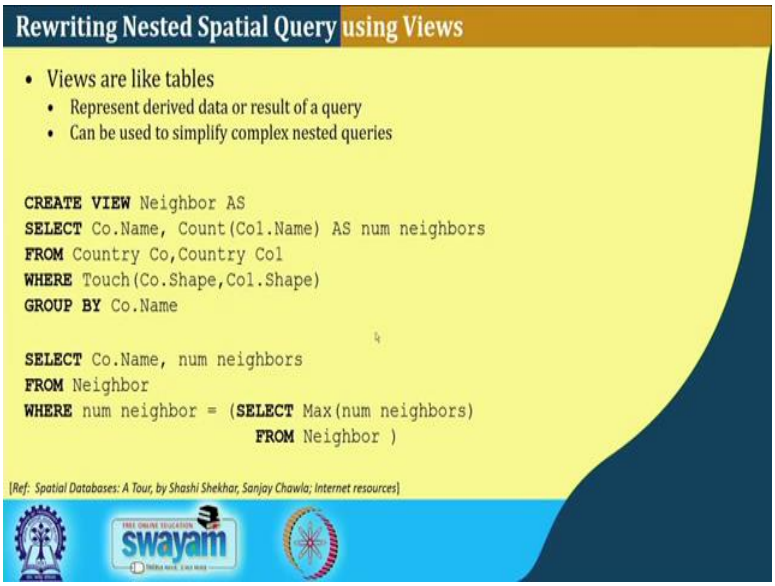
So, select from the country table Co dot Name, Co dot Name IN SELECT Co Name FROM Country Co and Country Co 1 again self-join touch Co Shape with the Co 1 Shape that two country touching each other GROUP BY the country name and there should be having a count of one. So, at least there are one it is touching the country right its shows a complex nested query with aggregate operations right. Such queries can be written in two expression namely a view definition. So, I can keep this as a view definition that initial thing as a view definition; that means, intermediate as you understand that a view table and the view definition in case of a standard database same concept out here.

Such queries can be written in to two expression mainly a view definition and a query on the and querying on the view. So, I create a view and then try to query on the view the

inner query become a view and the outer query in this case become a query run on the view or the query on the view.

So, I create a intermediate view and then query on the view actually thing. So, that is a nested query. This type of conceptually querying a view; a view definition or view table and querying they are also in our non spatial database, but here what we have this type of things like topological relationship like touch or in previously there other definitions.

(Refer Slide Time: 30:42)



So, what many of know, but those are; those are not immediately customed with.

So, views are like tables represent derived data or a result of a query. So, intermediate table I can say can be used simply by complex nested queries. So, if I have a complex nested query I can divide the I can go on creating means views at the at the lower level and then query on the things right like I can say create view neighbor at AS Co Name Count Name as the number of neighbors and country Co name and country Co1 Co and Co one touch Co shape and this. So, this is view it creates the country name, count at the major components which that the number of neighbors of the country.

Now, I use this view neighbor at as the country name and the num neighbors FROM the Neighbor is my view table and the num neighbor equal to select Max that is we want to select that the country having maximum neighbors. So, Max neighbors from the neighbor. So, neighbor is our is the a view definition.

So, what we tried in the today's discussion is try to see that if the; if the spatial query requirement or spatial SQL we require something more or something more dealing is required in order to have this in order to solve these queries. So, here what we have seen that there are different operations, spatial operations etcetera which are not straight forward, I cannot have those things in a standard query base like in this case touch, we have seen calculating distance, we have seen buffering operations and into the things.

So, these are the some of the spatial operations which we require for this spatial database consideration. So, let us conclude our discussion here and we will look at some of the more aspects of the SQL spatial SQL or the SQL dealing with the spatial databases in our subsequent lectures.

Thank you.