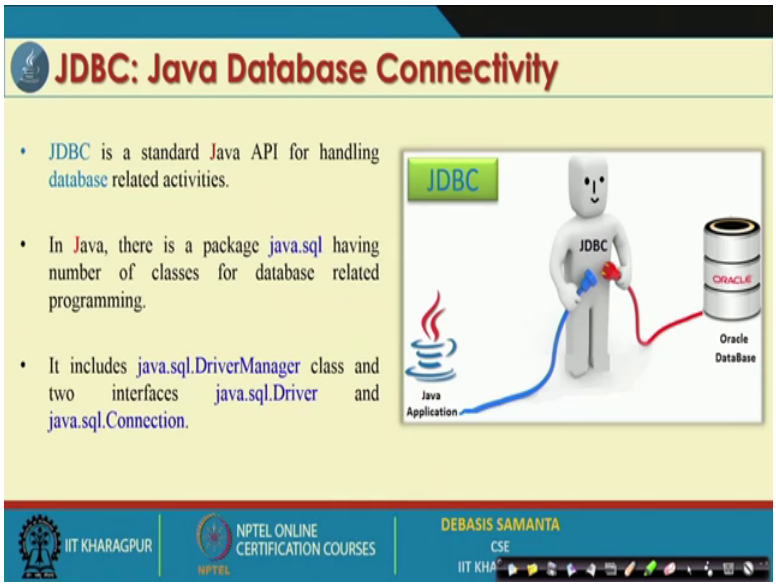
Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 51
JDBC - II

As we have already stated there, there are three things, so far the JDBC related java programming is concerned, the first thing is that the server which basically includes the database; that is the MySQL server we have covered, and the next thing is basically JDBC driver. So, this basically driver is a connector, connector between your program, your program written in java application to the database which is mentioned. So, there is a connector.

Now we are going to discuss about JDBC as a connector, so that it will connect a java programmer to the database. Now in the last examples what we have discussed about is a direct, there is no connection from the same host machine, where you have installed the MySQL, from there from the same console you have executed, but now we will basically see how from the java application in, as an end point we can communicate to the MySQL server remotely. So, this concept is called the java database connectivity.

(Refer Slide Time: 01:21)



Now let us see exactly concept first. So, as I told you java database connectivity insures you to have the connection from your program. This is your java application, you can

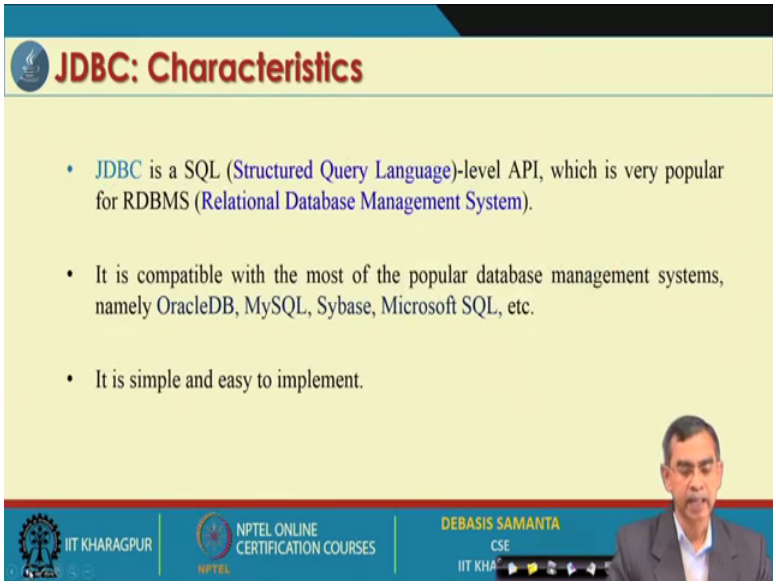
write whatever the program it is there using any API you can use it here. So, it basically is a program actually that you. Now your database can be stored here maybe Oracle or MySQL or some other system database is there, it is there. Now here is the java database connectivity JDBC driver, it basically connect your application to this database. So, this connection is a basically, is a connector line. Now for this connection you can in you can, ok fine. So, for the JDBC connections related API these concern, all those things are available in a package in java and this is a bundle package like API, so java dot sql. Anyway how to do this connection and everything to through JDBC and everything. So, basically JDBC related all the commands is basically, I mean implementation is available in java dot sql. We will discuss about java dot sql package later on.

Now, we will discuss about only the JDBC connectivity and regarding the JDBC driver. Now the driver that you can install from the driver is a java sql dot driver is a one package it is there. So, using this package, basically there are two classes driver manager and driver, three classes rather driver manager, driver and then this is the connection. So, these are the three things that you can have from this java sql dot package actually. So, these three classes are already defined there, we will discuss about all these classes later on. Anyhow in order to use the JDBC from your program definitely java dot sql should be imported in this application and then you have to create the object of these driver manager, driver and then connection. So, this will basically help you from your program side how the JDBC can be connected. Once the JDBC is connected with your application then JDBC will take care about the execution of any SQL statement that you used from your program there, and it will send this statement go to the server, server will execute and then result will be returned to the JDBC. JDBC interns return the result to your program, and then from the program you can have the result and process it and then you can get everything.

So, here basically directly if you can do it, but here through JDBC you can this one. And one important thing is that, this is basically is a object oriented concept, object oriented, paradigms it is there. Wherever this database which you usually use relational database. So, it is a relational concept, so paradigm is different. There may be some object oriented database also there, but usually a SQL follow the relational database, and JDBC is a unique one system we can say, who is basically match between the object oriented

concept to the relational concept. So, this is a very unique one features that the JDBC profile has and very useful also for any java programmers.

(Refer Slide Time: 04:39)



Anyway, so this is basically the about the JDBC database connectivity, and as you know JDBC it supports the SQL related query execution. So, JDBC in fact, is a SQL level API which is very popular for handling RDBMS; that means, Relational Database Management System.

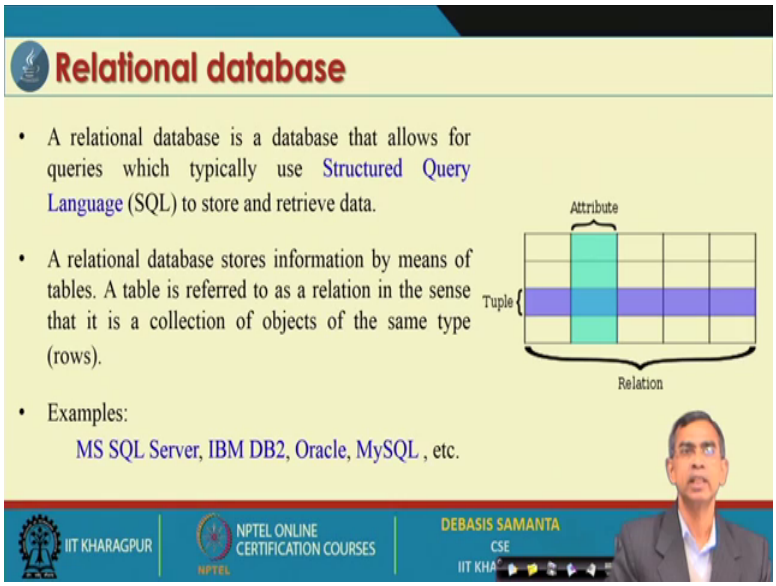
Now, there are some what is called the driver also available, who is basically allowed to handle with some other type of database management system also, but initially and till time right the SQL related RDBMS is very popular there. And there are many vendor those basically help you to maintain your database, as you have mentioned here for example, OracleDB from the oracle. MySQL is basically from the oracle itself, but it is a smaller version one, and OracleDB is very for the business purpose, for big business in the organization if they want to use it for this one and MySQL is for the small enterprises for organization.

Other than this one, one more popular is called the Sybase and there is also a MSSQL Microsoft from and that is also available these are the basically database, but in our, in this session, in this discussion we have used a MySQL. Basically syntax and everything are the same, you do not bother about whether it is OracleDB or MySQL or Sybase or MSSQL absolutely they are very transparent. You do not have to think about anything

which database management system has been maintained, but JDBC will be able to contact or interact any database vendor; OracleDB, MySQL, Sybase or whatever it is there, but here; obviously, for different vendor database, the JDBC driver is also different.

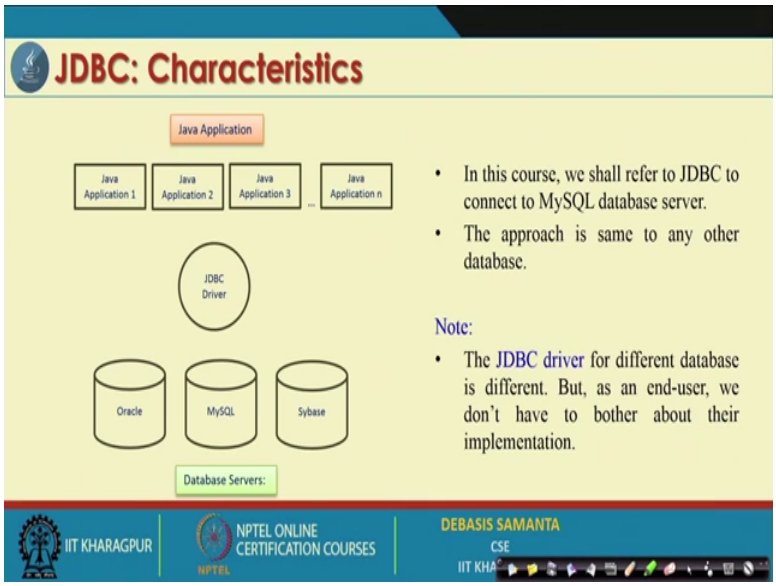
So, usually every vendor for example OracleDB create a JDBC driver for you if you want to use it. Similarly MySQL also create another JDBC driver for compatible to MySQL. So, you have to install that JDBC driver for which you want to have a connection. So, I will tell you exactly how the MySQL related JDBC driver can be installed in our machine.

(Refer Slide Time: 06:45)



Now relational database, as we already know that it is basically a relation database means is a table, a table contains a number of fields and also it contains a number of records, and that is why it is called the table. So, relational database means called tabular database, but it is called the relation, because the processing of the different entities or the records or the entire tables or many tables based on some mathematical formulation. This mathematical formulation is called relational algebra, as the entire database can be processed using the mathematics mathematical syntax, called the mathematical operations, relational algebra this is why this is called the relational database management system, and that is why RDBMS.

(Refer Slide Time: 07:38)

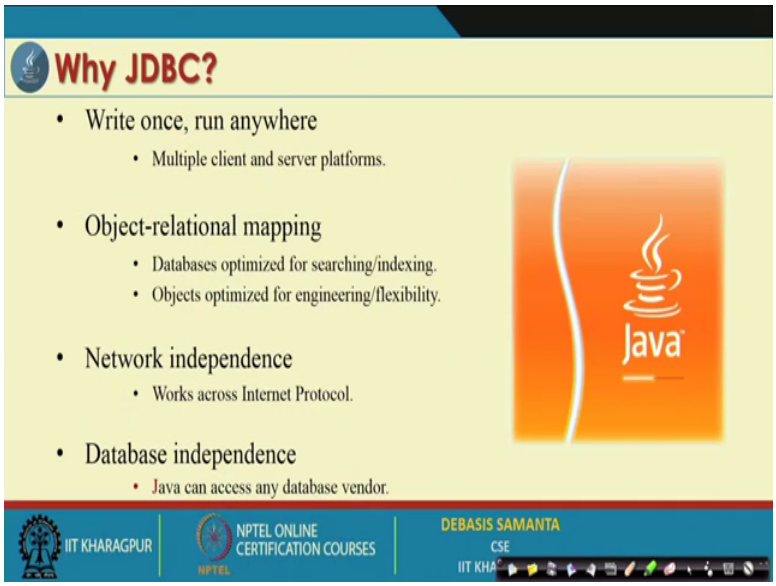


Now, so for the database driver is concerned which I was just talking about. So, if run your SQL statement from your application, as it will be run remotely to a server through JDBC driver and for different JDBC driver the different program is there. So, the different database management system; the different program is there as we see here.

So, this is basically java application any one application from anyone program right, so JDBC driver. Now as I say JDBC driver you have to install if you want to connect oracle for oracle compatible this one. So, from the oracle source only you can get it, install it. Similarly Sybase also you can get it, install it, MySQL you can get it, install it. And then this basically, so it will connect the database server and this is a Java application Java, JDBC will connect one application to the database management system through JDBC, that concept is basically, it is there, that is characteristic that you have to follow. I think it is easy to understand.

Now in our discussion whatever we will follow we will refer to the JDBC driver related to MySQL database server only. And the approach in fact, very similar to any other database oracle or Sybase or MySQL absolute no difficulties, only you have to install that appropriate driver in your system. The JDBC driver for different database is different as I told you but you have to just install this one.

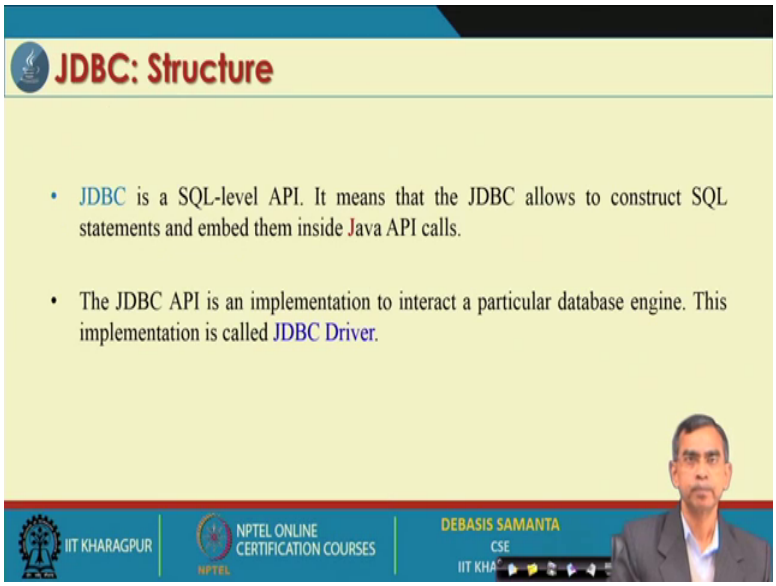
(Refer Slide Time: 09:07)



Now JDBC is popular, because it can be executed anywhere from any system. So, platform dependence independence is a very good issues, that mean as you know java application is a platform independent, so absolutely you can run this program from any system as you wish, and then it is a basically object relational mapping. I told you, so java application is object oriented, whereas, database is relation oriented. So, there is a good mapping for the JDBC and is a unique concept it is there. And network independence, in which network you belongs, absolutely you do not have to bother about it, JDBC driver itself will take care about all network related these things.

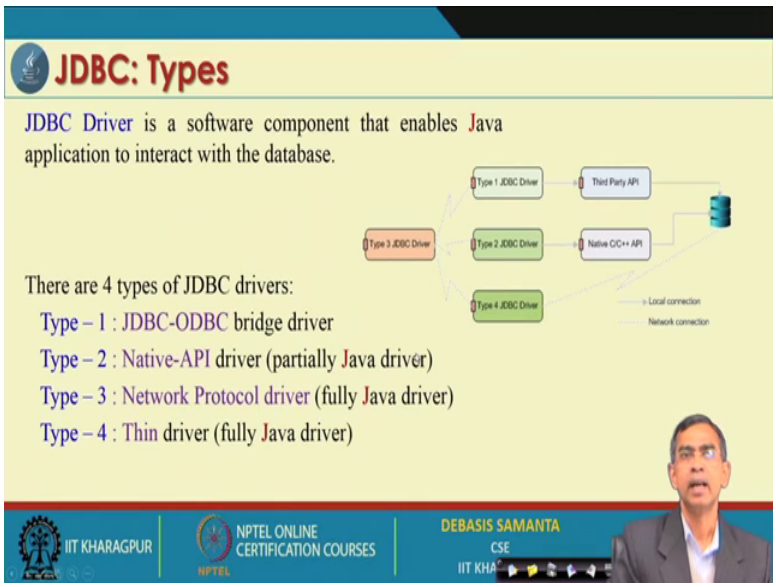
So, the network programming that we have learnt about, this is for the core related to some other network protocol implementation. Whereas the database connectivity also the through network only, because your remote server can be maintained in some distance location, but JDBC will connect from your application to database server and both to and fro connection. That means, if you; if the database server return some results, the results will be collected by the JDBC driver and it will return to your application and vice versa. So, this is the database independency is a very important. And database independency is because, as I told you it is also applicable for relational database, but there are some systems for which a, JDBC driver is also available which basically allow you to connect it, for object oriented database or some other database management system.

(Refer Slide Time: 10:37)



Now, they, as you have told you, so in order to use the JDBC, definitely you should know SQL, because you have to fire the SQL related query statement through JDBC driver. Now, I will just discuss quickly, that there are different type of JDBC driver. And in this slide as you see, there are four different type of JDBC drivers are there.

(Refer Slide Time: 10:57)



So, the first type on driver is called the JDBC-ODBC bridge driver. So, it is basically object oriented database corporation, is basically the idea about this JDBC driver is that, so they can use any third party API; that means, not the java API. So, if these driver you

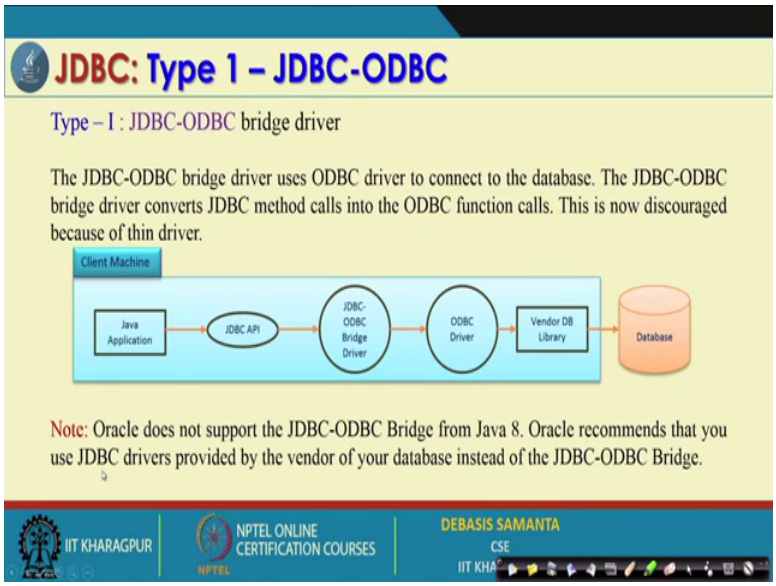
access the third party API then it is called the type 1 JDBC driver, and usually no network connection they support. So, if you install JDBC driver in your host machine, and in same machine the third API is also installed, then you can use the database from there.

So, it is not the, remote connection is not possible. And next is the type 2 JDBC driver there are some Native-API, so you can develop your own API or already existing C, or C plus plus libraries; if we use it then this driver can be the type 2 JDBC driver, so that kind of driver is also available in the market. However, the same type 1 driver like this also connect to the server in the same host, no network connection is possible.

On the other hand type 3 JDBC driver, it can consider any one type 1 or type 2; however, connected through net. So, this is basically network connection is there. So, this driver will connect the network, connect to the remote one host, that host can include the JDBC. That kind of concept is there, so this is the type 3. And finally, type 4 JDBC driver is a basically independent, it can be independently connect to the network and then you can use it. So, this is basically is a good for remote server maintenance and other issues are there.

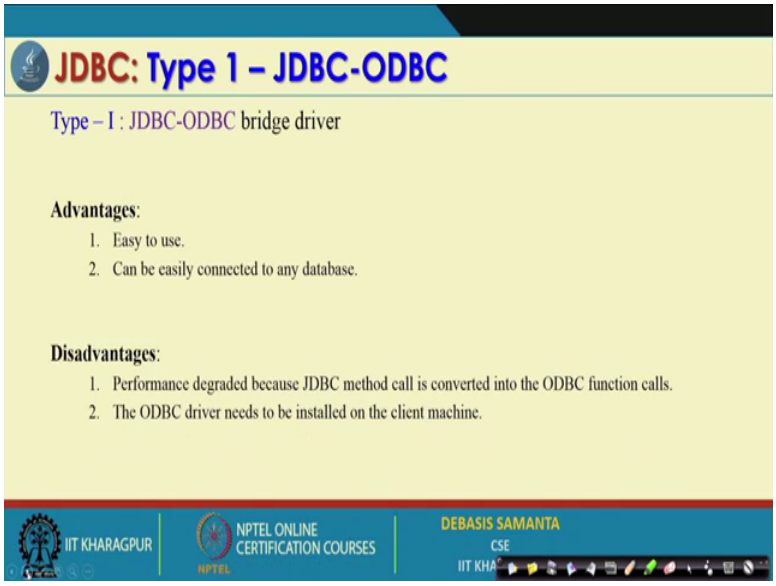
So, whatever it is there, we will basically follow the type 4 JDBC driver in our discussion. And as you see the type 1 is a bridge driver, it basically bridge from this java API to other API like. Whereas API driver, the type 2 driver is a partially java driver and then type three and type four is fully developed by java system only. So, they are the hybrid actually you can say, these are the two hybrid JDBC driver actually. In some situation if you want to have cross platform application then you can follow it.

(Refer Slide Time: 13:24)



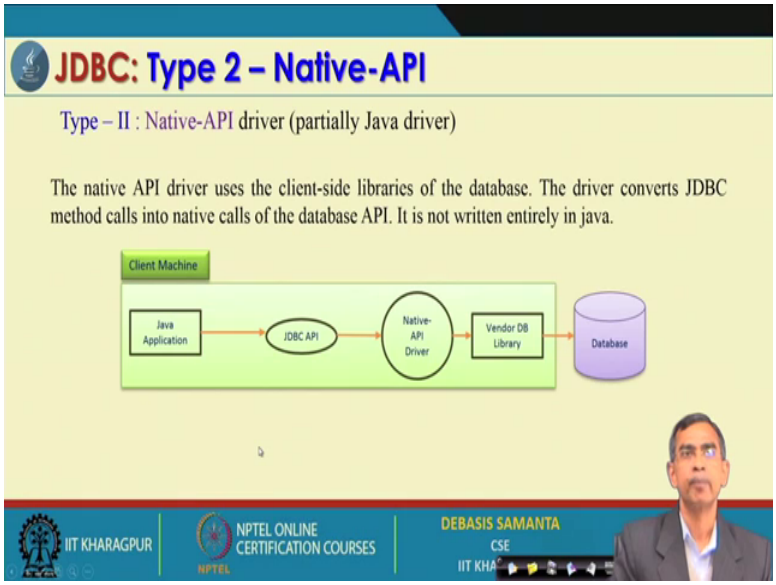
So, these are the driver that is there, and then Now, how the JDBC-ODBC work it is, basically this is explained here, these are database as I told you, everything should be installed in the same machine that is the client machine and if it is your job application you can just in the same machine, host it and then the same machine also host the database management system. So, this way it basically works there, but if this data; this type of driver has certain pros and cons.

(Refer Slide Time: 13:51)



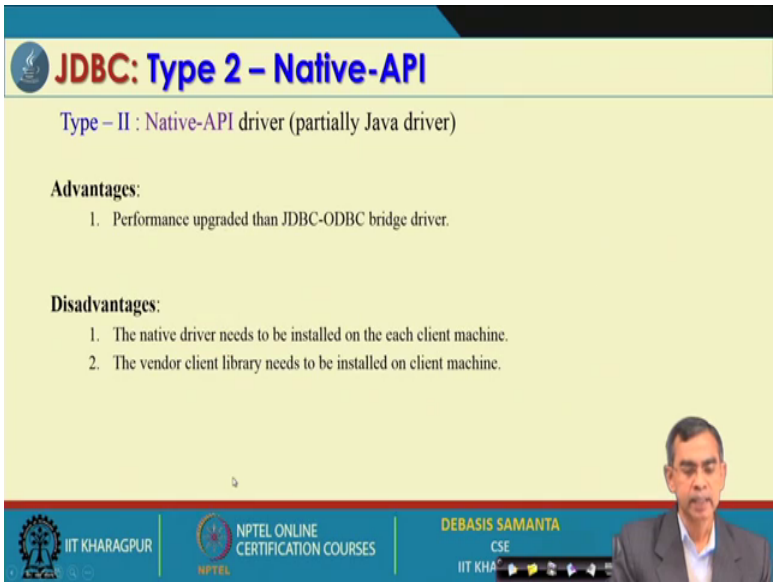
Advantages is that it is easy to use and can be easily connected to any database, that is a good advantage; however, its performance has an issue, because it, the performance degraded because JDBC method call is converted into ODBC function call. So, is that query execution is not so fast, and then the ODBC driver needs to be installed in the same client machine. That means, both JDBC and ODBC needs to be installed

(Refer Slide Time: 14:18)



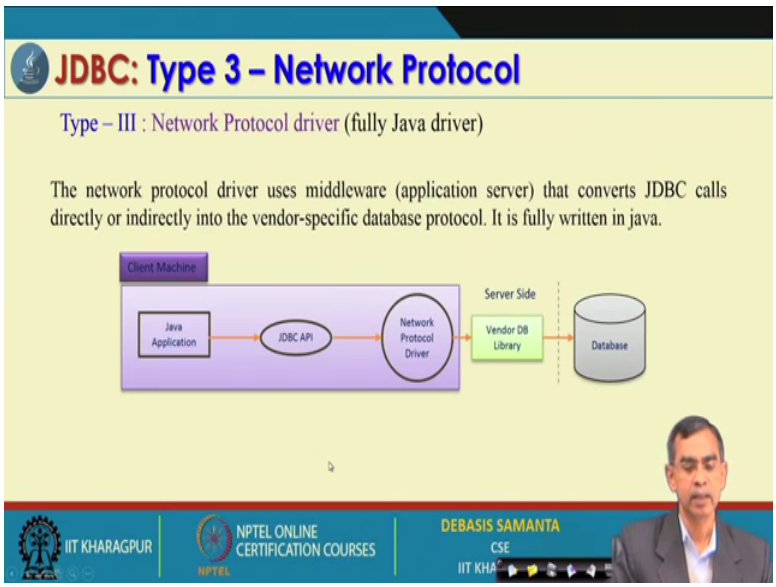
And then next is type 2 native-API; that means, as I told you, you can connect to your own API or some already existing API that those are there in C C plus plus like. So, here is a same thing just like type 2, in the client machine you have to install all these. So, JDBC API should be installed and then that the database also needs to be installed, but only it will connect other than other database, it will connect to the native or C C plus plus API is there and this is your application. So, these are different facilities of service that it can give to you.

(Refer Slide Time: 14:55)



And our next example, and then it has some; obviously, advantage and disadvantage. It is also again performance issue also it has, because it needs to be converted from your API to other native API like, the native driver needs to be installed on each client machine as the same issues it is there. The vendor library needs to be installed in the client machine also, because JDBC driver also to be installed and native API is those you have to use it, also needs to be installed in your same machine. So, these are the things are there in this particular example

(Refer Slide Time: 15:32)

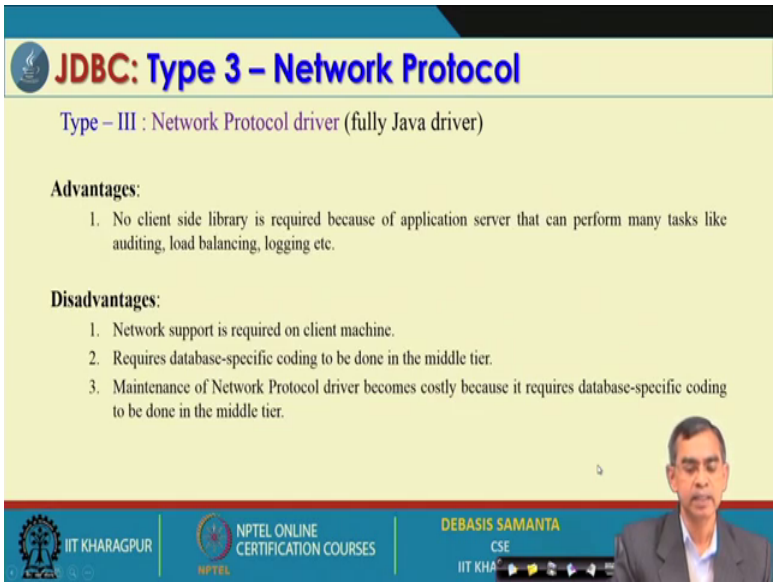


And type 3 is basically network protocol. As you know type 3 database can be connected through net, whether is a type 1 or type 2 it also can be connected, but usually it is type 1 independently, and here is the idea it is. So, basically these are the server site that can be in a remote location, connected through the network. And then network protocol driver, they are basically you will connect it, that needs to be installed in your machine of course. And this basically the JDBC driver, JDBC driver we will follow what type of network protocol that needs to be there. Actually JDBC driver itself include this protocol bundle with this one.

So, you do not have to explicitly considered both the things will be there. Only in your machine you have to install this JDBC driver. And then finally your application, java application using your JDK and whatever it is there you install it. So, you can run your java program and then get a connection through JDBC, JDBC will send the connection to the database server, get the result, result will be return.

So, this is basically the type three. Type three is the most one sophisticated things it is there; however, there is another type 4 database is also there, but now let us see what is the issue, goodness good point and bad point.

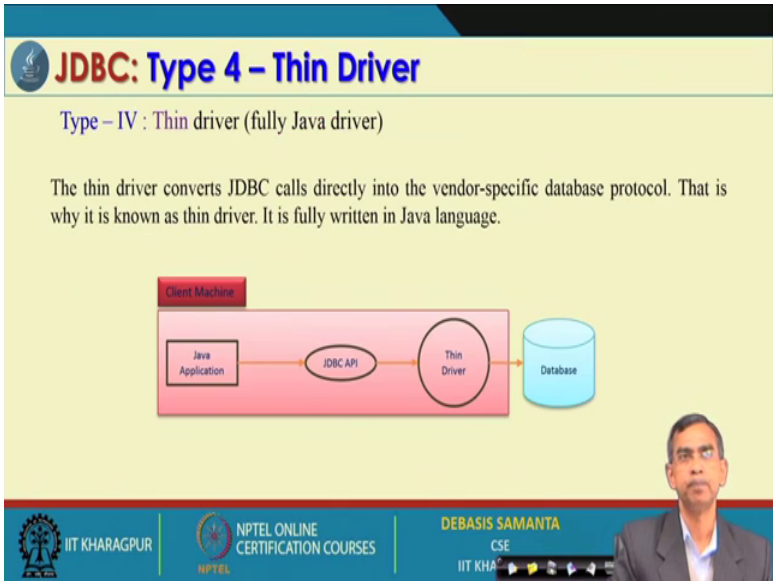
(Refer Slide Time: 16:44)



So, for this type 3 is there. Here the no client side libraries required, because JDBC driver is installed in your own machines only, only the JDBC driver and then your java needs to be installed there. And disadvantage is the network support is required, because

without network and communication you cannot connect to that, and it required for database specific coding to be done in the middle tier. So, the database coaching means it will already database JDBC will take care, but some coding effort is required time to time. And final is a as it is related to network, so network protocol needs to be compatible to your system.

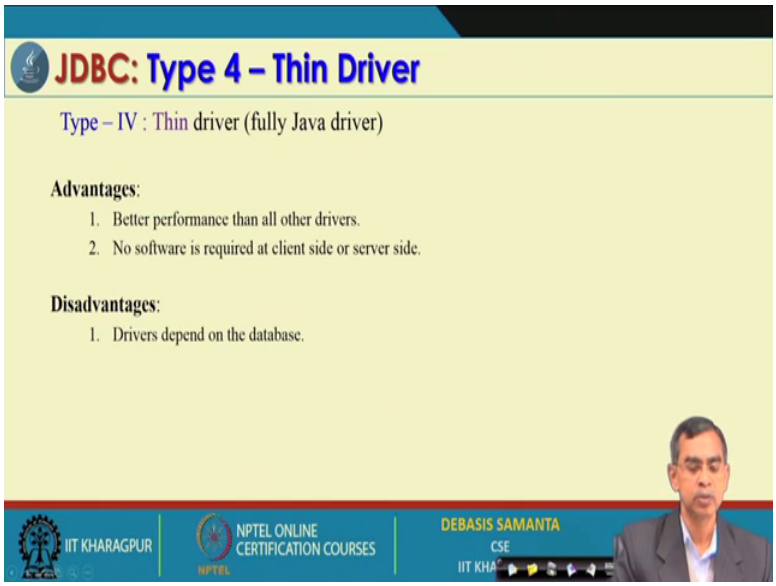
(Refer Slide Time: 17:23)



Now, type 4 actually is an process to eliminate all those things is there, it is more sophisticated than type 3 here.

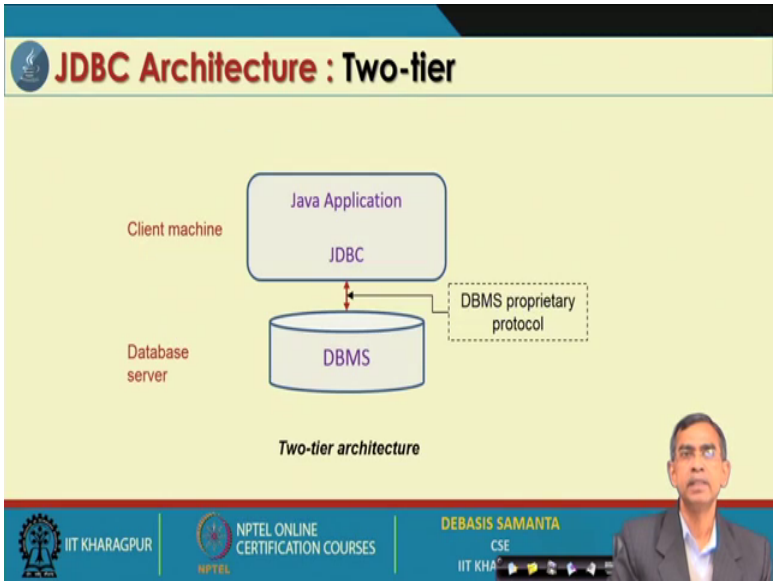
So, here again database is in a remote location and thin driver it is basically called, which basically take care about the network and then connectivity issues, between your JDBC java application duo and it will do this thing. So, it is more sophisticated and as it is, again few issues are there like say type 3 database issues are there

(Refer Slide Time: 17:53)



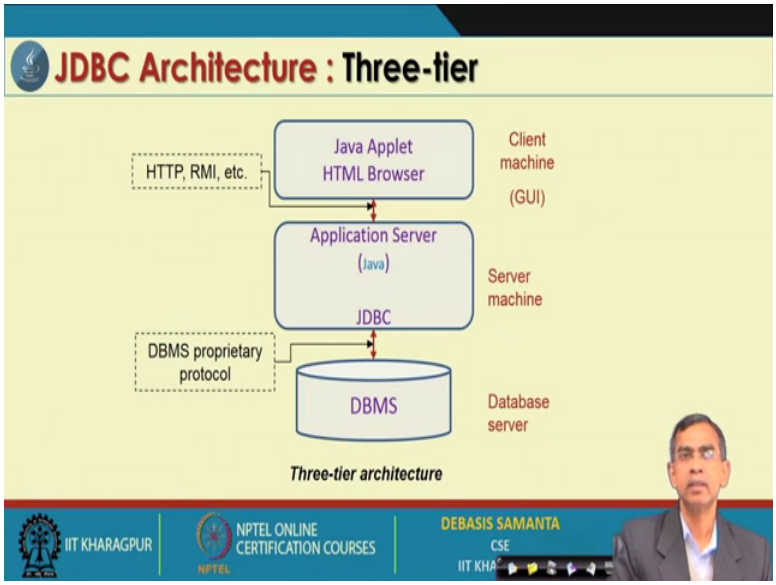
Thin driver is required, as you know basically, and better performance than all the drivers that you have considered is very fast, and no software is required at the client side or server side as well as. And here the drivers basically, different drivers from the different vendors are be installed. I think this is not also big issue we can do it. So, we will consider the type 4 JDBC driver in our in our discussion. Anyway, so we have learned about JDBC and JDBC characteristics,

(Refer Slide Time: 18:25)



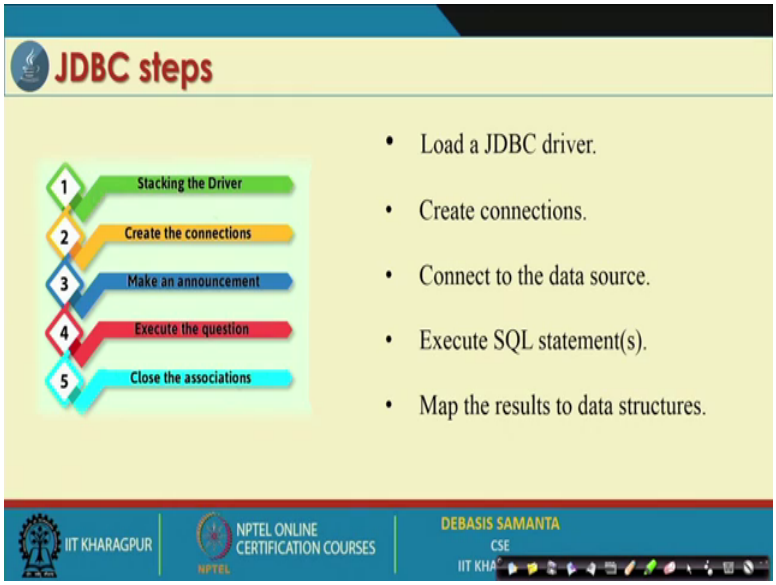
Architecture maybe two tier architecture, all the type 1, type 2, all these things are the two tier architecture.

(Refer Slide Time: 18:34)



And then three tier architecture is also similarly type 3, type 4, JDBC driver are the three tier architecture.

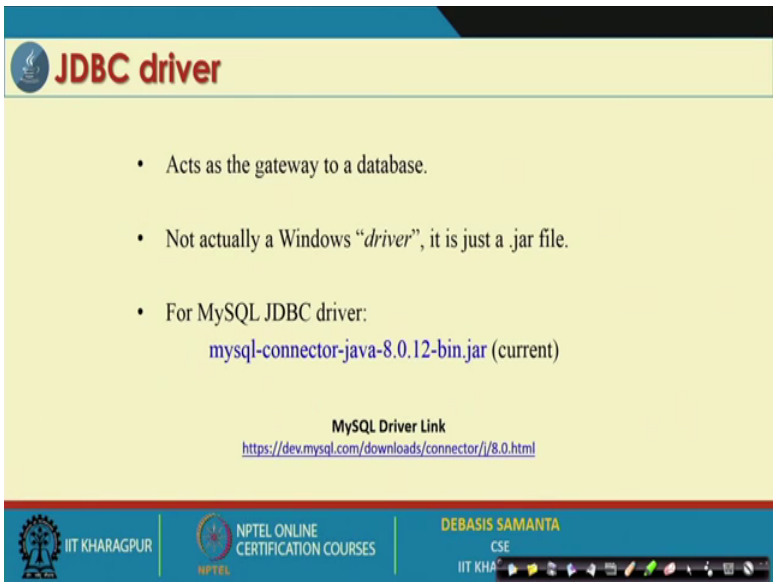
(Refer Slide Time: 18:44)



Now, let us see what exactly inside the JDBC is there. Now, so there are few steps actually if you want to invoke the JDBC in your application. So, first you have to load a JDBC driver that can be done from your program itself by making some classes which

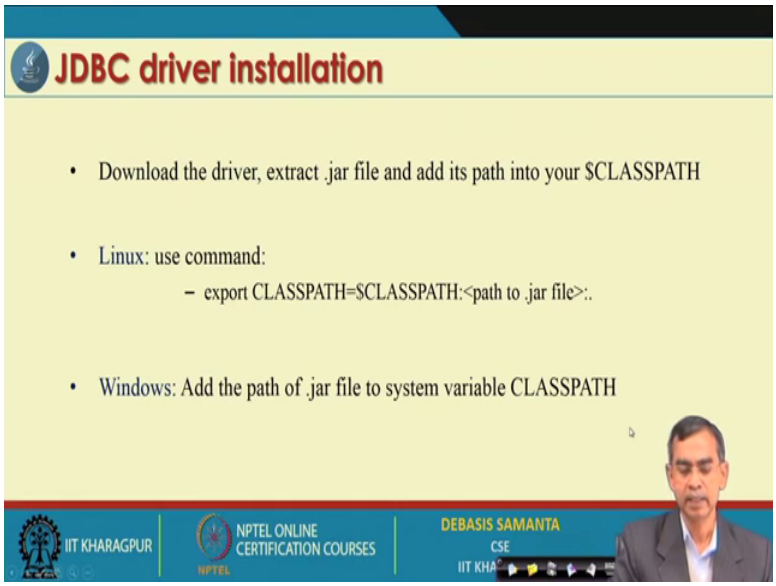
basically provides you to how to load a driver to your application, and then you have to create a connection. There is again class facilities are there, and once it is connection is there you have to connect to the data source, which database you want to connect it and then you execute SQL statement, that way you have already explained during our MySQL studies and then the result that you will receive you have to process the result according to your own requirement. So, these are the five step procedure that we have to follow and we will learn exactly all this five procedures, how it can be executed carried out in our, from the java application

(Refer Slide Time: 19:32)



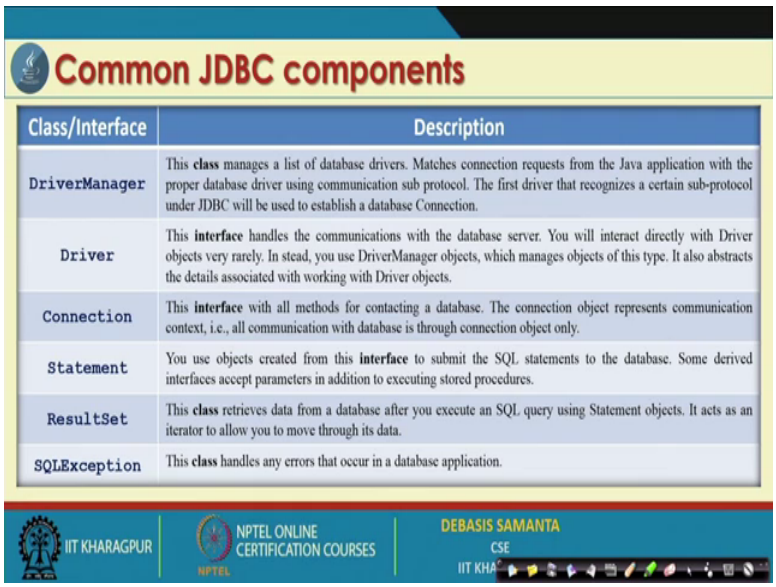
Now, so far the JDBC driver is concerned, we have to install this driver. This is the one link I have given it for you, so you can go to this link and directly installed it, and this is the most current version it is available. There is also other link is also available, from here also you can download this, this JDBC driver that is in your case, that the type four JDBC driver actually we have mentioned you. Once this installation, that that installation is very stress for, you just download this one and double click it will automatically installed in your machine. We have to then finally, set the class path and all those settings are minimum requirement that is required and then class path setting.

(Refer Slide Time: 20:10)



If you know a little bit system level administrative job in windows, you can do it otherwise in our next demonstration, we will give you the details about how the class path can be set and then connecting to database I will come to there, how the connector connection can be done.

(Refer Slide Time: 20:25)

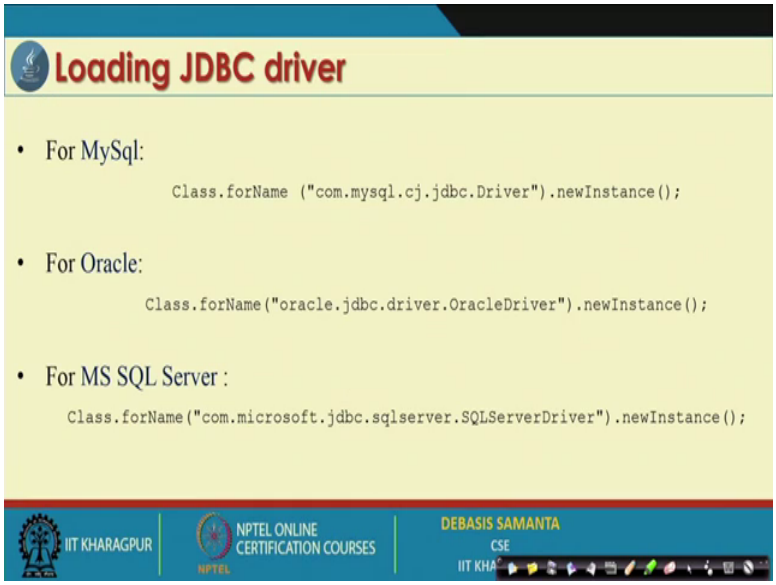


Now, I told you that all the things can be possible from your application itself, so you should follow some API. So, the API that is responsible for doing this thing is java dot sql API, that is a package. So, you have to import java dot sql dot star; that means, this

package contents the different classes as we have mentioned here; DriverManager, Driver, Connection, Statement, ResultSet and SQLException. So, all these classes are enough, or only few these are the sixdifferent class, libraries are there in your java dot sql API.

So, learning all these class is basically sufficient to use it. So, we will try to learn all these classes in our next, in this current session as well as in your next session, so that how we can utilize this one. DriverManager is responsible for maintaining the driver and driver is basically, the controlling the things, and then here basically the driver manager statement, and DriverMmanager results its and then SQLException are the classes, whereas Driver, Connection and Statement are the interface; that mean class facilities are there and we have to use the interface to redefine your requirement actually, we will discuss all these things.

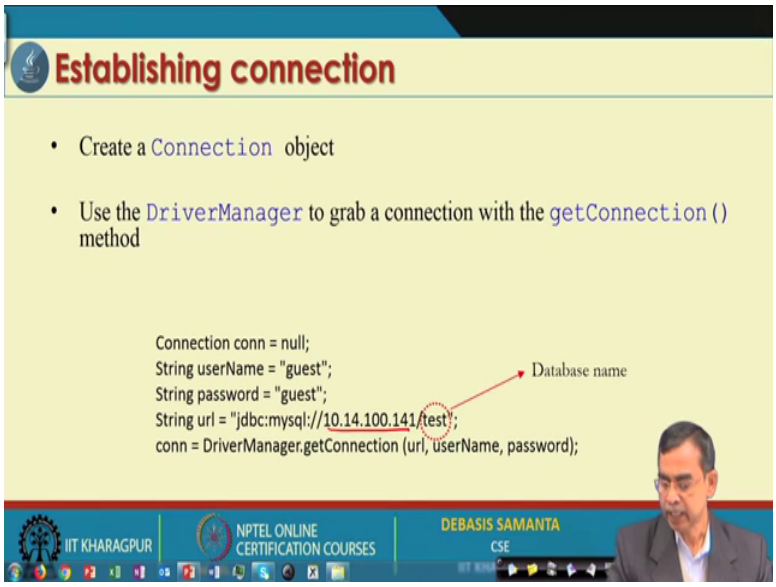
(Refer Slide Time: 21:32)



Now so far the loading the JDBC driver is concerned from your program, you have to issue this statement. So, from your java program if you execute this one, and here is basically your type 4 driver, where it is basically it mention. So, you have to give this one. And then these basically, your program will connect connect to the JDBC driver automatically and then it will basically establish a connection, it is very simple. On the other hand, if you want to connect the Oracle, this is the driver that loading procedure, and if it is for the MySQL, this is a loading procedure. For the different what is called the

database management system, the different loading procedures are there. So, that information also we can have it from the vendor itself, so they will tell you that how to load your JDBC driver to connect to that particular database is there.

(Refer Slide Time: 22:28)

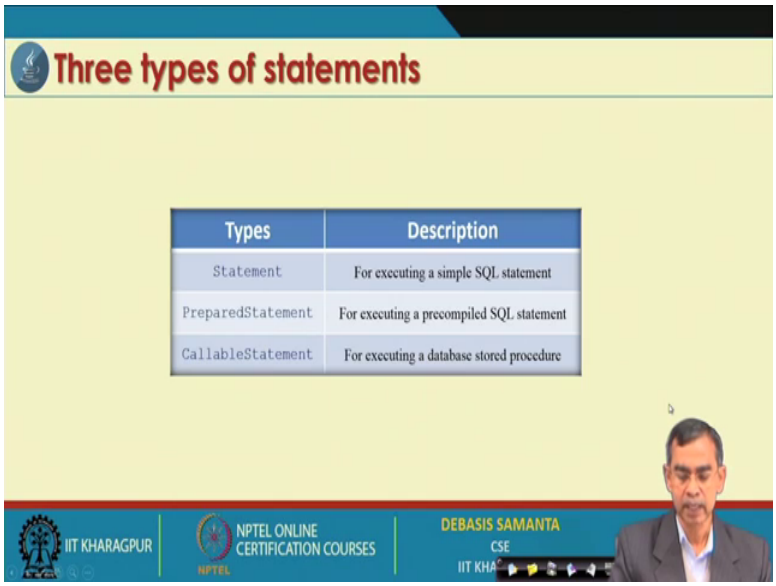


And then final the Connection object is there. So, you have to create a connection object as I told you, connection is a class, is an interface, which is already there in java dot sql package. And then use the DriverManager, this is the class one and there is a method getConnection. Now, here is an few syntax that I have told you, so you have to create a connection object. This is the connection object conn; initially it is null and then this is basically userName “guest” and then password “guest”. Suppose, that the username and password always, it should be specified. In our earlier discussion about MySQL, we have given the username as root and password as root also. Otherwise, you can say it whenever you install the system, that time you can give the username and password and password can be changed also, anyway. So, suppose username and password are the two objects; two string is “guest” and “guest” like you have just have it and then this is basically url one string that basically in which machine your, I mean server is there. Here is basically, if you see, your SQL server if is installed in this IP address then you have to mention it, and then in which directory that you have to mention it and this is the name of the database that you have to do it. For example, “test” is the database, it is installed in this remote server whose IP address is there and this is basically the directive, where it is stored there. So, this url actually the, for any connection, as you know in Java

networking we have told you, url is basically a specific location of a particular server where it is there. and once this url is already known then you have to create the connection, the connection object is created here and then it can be instantiated by calling this method, DriverManager. getConnection as I told you and passing these are the three arguments url, userName, password.

So, this is the idea about how you can connect to your application to the remote server using this statement. Now, here, so the idea it is, so, Java MySQL database is basically name of the driver, where it is loaded and this basically, database server where the location and finally, this is the name of the database, those three things are very important. Now, again if you want to install your server in locally that is also possible. Means, your JDBC driver and your application and the server, database server both are in the same machine. Then here in this case you should write local host, that is all. So, this will completes the idea about establishing a connection from your application to the database server.

(Refer Slide Time: 25:13)

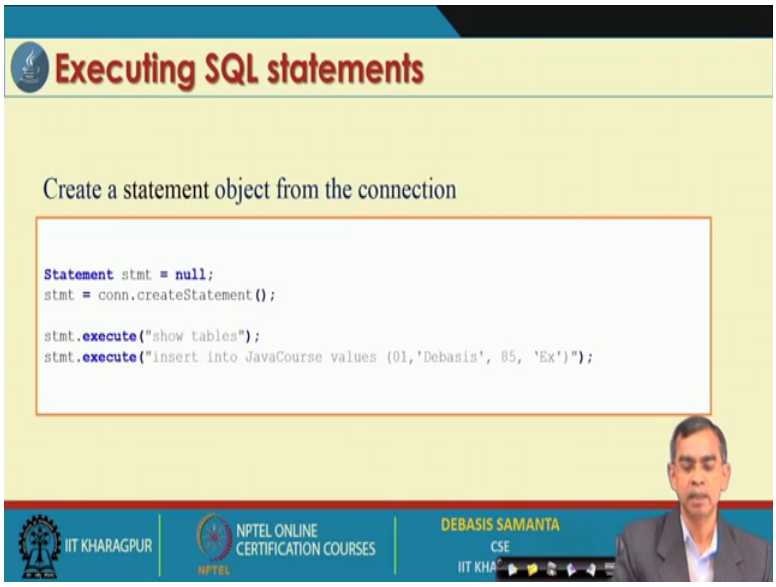


Now, so far the statement execution is concerned in JDBC, in java API; java dot sql API. There are one class; statement is there.

Now, this class has the three different predefined classes are there called the Statement, reparedStatement, and CallableStatement. So, Statement is basically executing one statement at a time. PreparedStatement is set of statement can be punched together to

these things one, and CallableStatement means; if you want to change the statement execution, when you are issuing this one. Anyway Statement and PreparedStatement, if you want to execute one statement or multiple statement at a time.

(Refer Slide Time: 25:53)

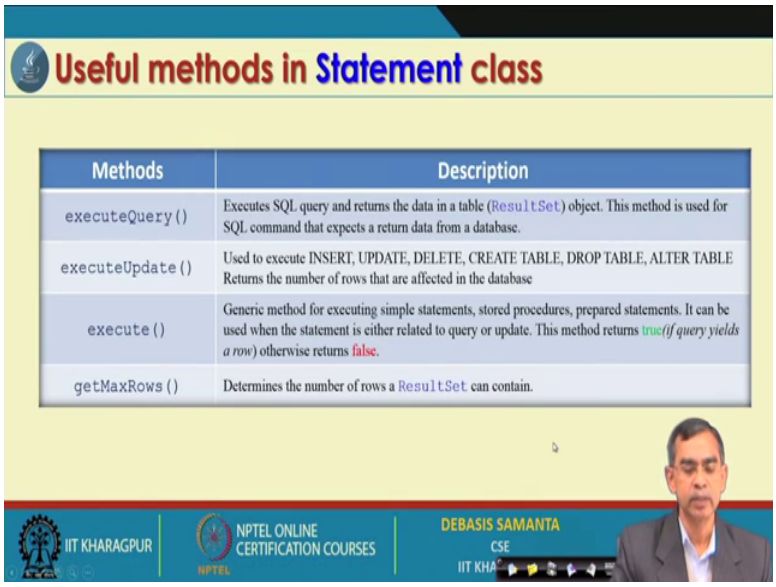


Now, here is a simple example that we are going to discuss about, how we can execute some SQL statement using our JDBC data connectivity already there from our program. Now, in this program, we have to input these are the statement. So, Statement is an object, is created. Object is the type of the class statement initially, it is null, so the object is just defined. And then the connection, with the earlier discussion, if you create a connection to this one so, connection, this is an object, connection object. And then the CreateStatement is basically is instance, the statement object. So, this statement and connection are very important in this case.

Now, once the statement object is created using this one successfully, if there is no error, then you are free to execute any statement. All the statement you can type in English, whether lowercase, uppercase, it is not an issue. Then there should be enclosed in double quote. For example, I want to execute “show tables” or show databases, then I write show databases or show tables within double quote. And then execute method can be called for, thus for the statement object, so it will basically, so you and once this is there in your control, all the database, all the tables which are there will be displayed into the system. Similarly, another statement “insert into JavaCourse value” this one.

So, as you have already learnt about how all this statement we have executed from the same concept, but here from the Java application, it will go to the JDBC driver, JDBC driver will send this statement to the database server, database server will execute it, and then you can get the result also. So, this basically tell you how the different statement can be executed from your JDBC, from your java application. There are many other statement also like wise we can execute it.

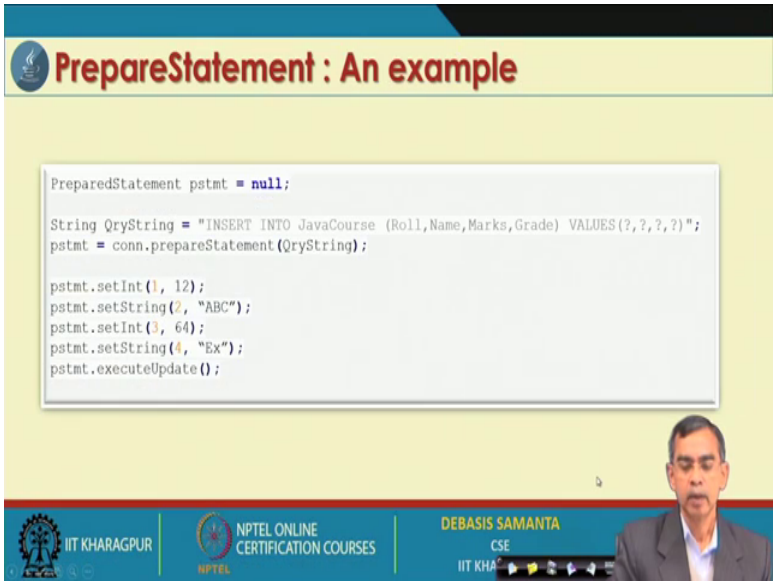
(Refer Slide Time: 27:48)



Now there are; obviously, few more useful methods as we have used one execute method. Like execute method there are some other method also there they are basically same thing right, whether it return something or not, so executeQuery, executeUpdate, Execute and getMaxRows. GetMaxiRows means, it will basically tell in a data that how many method, how this statement if it is executed, how many methods, how many results have been changed or updated like this one. On the other executeQuery it is basically creates a data, what is called the return, it basically return a ResultSet actually, the ResultSet regarding we will discuss in details. Basically if you execute a query it will return some results. So, those results can be captured by one object call the return; object of ResultSet class, and then it can be stored there and it basically executeUpdate, means it is update it and then finally, return Boolean that means, whether true or false like this one execute also execute a same way, but it is basically the different methods are there, basically the same thing whatever it is there.

Now, execute can be used without any result set. Whereas, these can be used with result set or simple the report that report during the execution line. Anyways, so these are the different statements that is possible for the statement object, and we will use in our demo as well. In the next session also, we will discuss about, details about the programming, so that how the results can be processed.

(Refer Slide Time: 29:25)



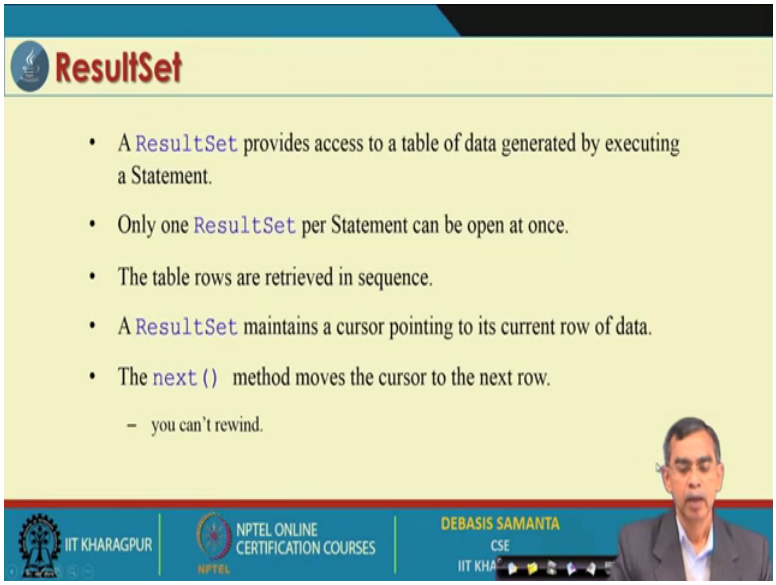
Now, this is an another example, called the PrepareStatement, as we see here, this statement is basically we can create a QryString. It is basically statement “INSERT INTO JavaCourse”. This is the one statement and that value actually, we have not mentioned it here, but that value can be given it after this query string is prepared we can push this value by means of prepare statement. So, prepare statement is another type of statement.

So, basically pstmt, setInt and then 1, 1 means the first value with this value 12 , 2 means the second value with this 1, 3 means the 3 value 64, 4 is this one this way. So, this is basically first is basically argument indicates that, which values and this is basically what is the value for this to be by which this one.

Now, what you can do is that we can read all this things from the keyboard or some other means by input output system or whatever it is there, then we can load all this thing into the statement, and finally, we can execute this one pstmt executeUpdate; that means, it will fire this statement finally.

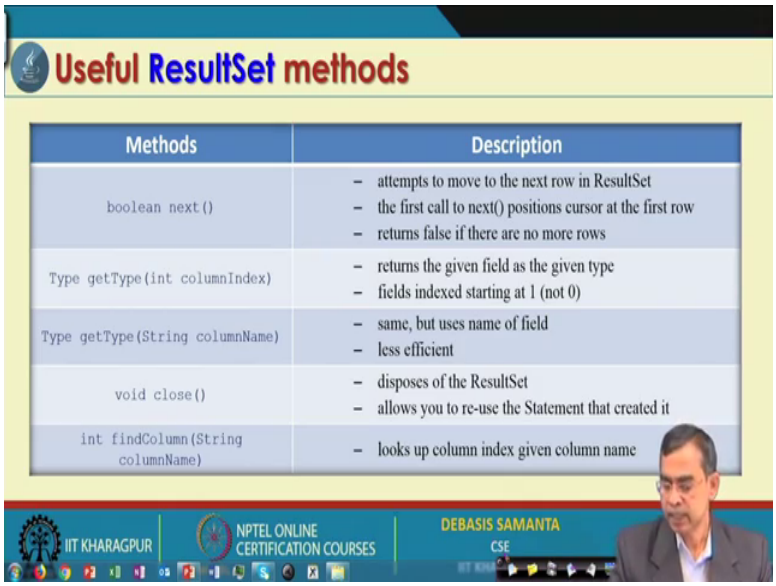
So, this is the idea about; that means, CPU want to prepare one statement and then we can use the prepare statement object for that. So, this is the prepare statement class.

(Refer Slide Time: 30:42)



And ResultSet is basically as I told you, it will store the results written by the by the execution of a SQL statement through the java application and so, ResultSet is basically is a complex on concepts, but it has, it returns you number of rows basically, if you execute a statement, which can returns a number of rows like, so if the number of rows are there, you can process or you can scan from one result to another result by means of next method, it is there. So, next method is there. Any way, we will have, let us have some examples so that we can understand about ResultSet methods are there.

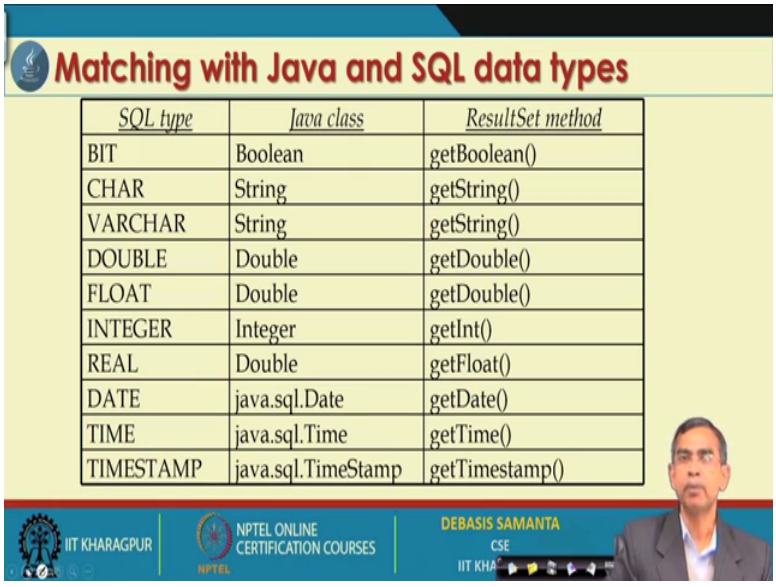
(Refer Slide Time: 31:20)



So, for the ResultSet class is concerned, these are the different methods, we have discussed about it. So, better all this method will be understand during the application.

In our next session, we will discuss about different application of this result sets one by one.

(Refer Slide Time: 31:38)

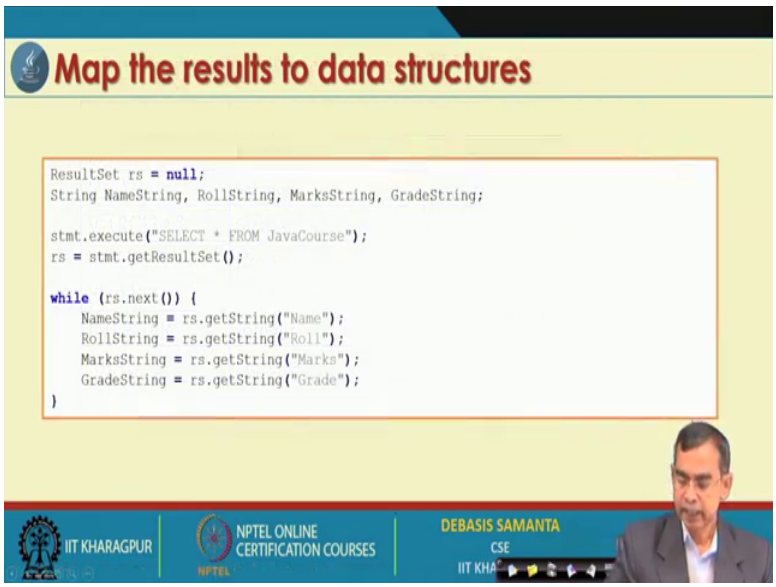


And there is again another important aspect that you should remember that different data type. See SQL also provides certain data type, where as java provide some other data type. So, there are mismatch in fact. So, here we have mentioned about. So, Boolean if it

is the data type that sql language can understand, SQL there are structured query language can understand. Whereas Boolean is in java, it is like a Boolean class. similarly, character that is there in SQL, It basically understand String and VARCHAR is basically string of characters, it is String, DOUBLE is Double, FLOAT is Double, INTEGER is Integer, REAL is Double, DATE is, this is the class Date, TIME is class Time, TIMESTAMP that is followed by this one, this one.

So, you have to just matching, whenever the result set that can be obtained, so they can have this kind of data type from the SQL side. And whenever you receive from your application side this is the data type, so there is a conversion is possible. So, there is a method of this conversion. So, these are the method by which the data type from Boolean from the SQL to the java program Boolean and then these are the method actually get method, it is called most popularly called the ResultSet get method. So, by this method you can convert the data easily into your java compatible things are there.

(Refer Slide Time: 32:56)



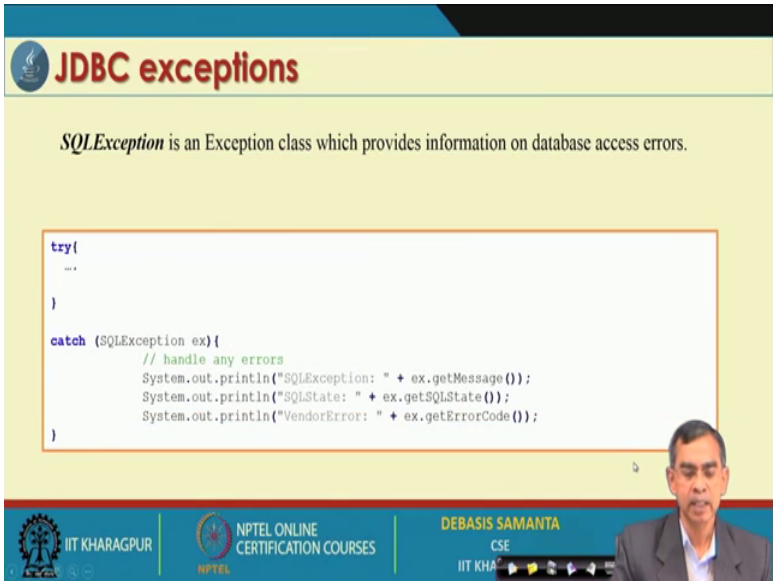
Now here is an example let us see, how the result set can be processed here. Assuming that the connection is maintained and then JDBC driver has been loaded and a connection is established, and now we are executing the statement and then this is suppose, this is the statement that we have executed. This is the statement suppose, you have executed, the “SELECT star from JavaCourse”. So, this means as you know, if the

database includes; contains certain record, it will give all the records that is there in the table. So, all the record will be stored into this rs object.

So, rs object is basically, the entire records, the entire rows basically, contains. Now, here while rs dot next means this is the result set. We will content, this is a record one, then record two, record three and so on, so dot n records are there. So, rs dot next basically, when it is not n, n means end of this records, it will process it. Name string rs get string; that means "Name", which is basically the first field will get into these one, "Roll" get into this one, "Marks" get into this one, "Grade" get into this one.

So, get string method is basically, get whatever the type it is there in the string and from there will be store in the string format. And from this string format if you can convert in the integer or double or whatever the type that is required there, so that you can get it. So, this way the one by one each record can be processed and it can be used in your purpose. So, this is the idea about the results returned by the JDBC connection and then it can be used there.

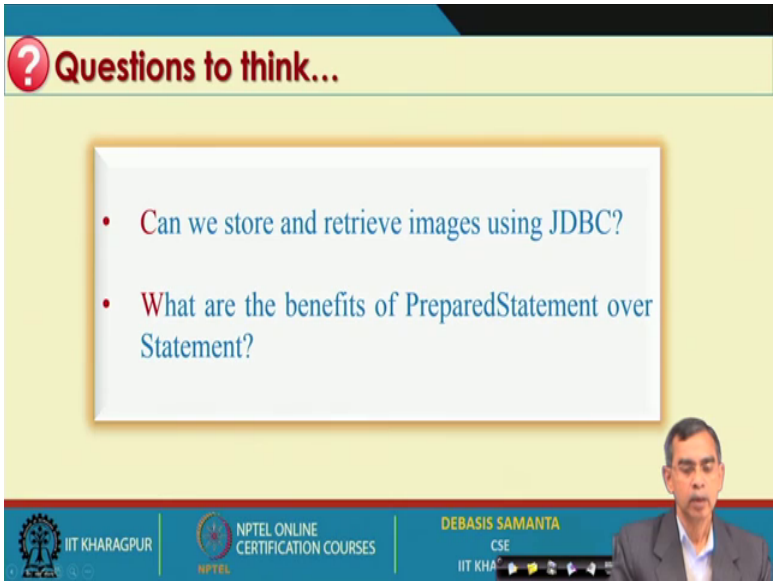
(Refer Slide Time: 34:35)



And another important thing is that exception is very much importance, so far the JDBC is concerned there. So, lot of try catch block can be added there, because in exception may occurs many times, say suppose you are not able to establish a connection so error will occur. Suppose, you have connected the establish, but because of the network disruption sometime the data is not accessible.

So, in order to avoid all those kind of unwanted disruption in the execution, so all those things can be caught by exception classes. So, the JDBC in the API, JDBC API there is a lot of SQLException is there and few exception methods it is there SQLException, SQL State, SQL error all those things can be getMessage, getSQLstate, get SQL error code, so many things are there so that knowing the, what type of error you can process it. So, these are the essentially exception handling mechanism needs to be followed in this case also; those things every statement which you are going to execute or every statement there in your program, better can be put under the Java exception handling mechanism, these are the (Refer Time: 35:50).

(Refer Slide Time: 35:51)



So, we have discussed about JDBC driver, connection and then establishing and then execution of the statement and then once the result is there. In our next session, we will discuss about more how to execute statement, all the connection, all the five steps that we have already told you regarding the JDBC application, we will discuss about all this five statement, how can be executed and then how the different execution can be carried out. Now, we can store and retrieve image using a JDBC or not. The answer is yes, but the procedure is bit difficult. Although ODBC is very good for storing what type of data, but sql is basically good for only storing the other, but that storing is not directly it is possible, some indirect mechanisms are there. And then prepared statement we have used it, it purpose is that, if you want to get the data one by one, those data can be collected in

your application and finally, create a statement to be executed then the prepared statement is there. So, these are the obvious questions, probably you have learned it fine.

So, thank you very much.