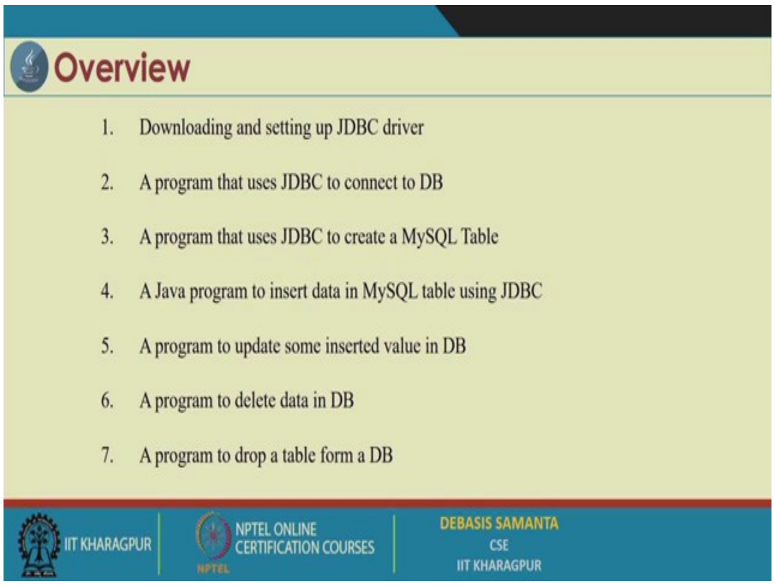
Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 54
Demonstration - XXI

There are three things so far the JDBC is concerned; one is the database management system which is basically in this learning we have considered MySQL and then another is that JDBC driver which is basically you will connect to the server and then third component is basically the Java application.

(Refer Slide Time: 00:31)



Overview

1. Downloading and setting up JDBC driver
2. A program that uses JDBC to connect to DB
3. A program that uses JDBC to create a MySQL Table
4. A Java program to insert data in MySQL table using JDBC
5. A program to update some inserted value in DB
6. A program to delete data in DB
7. A program to drop a table form a DB

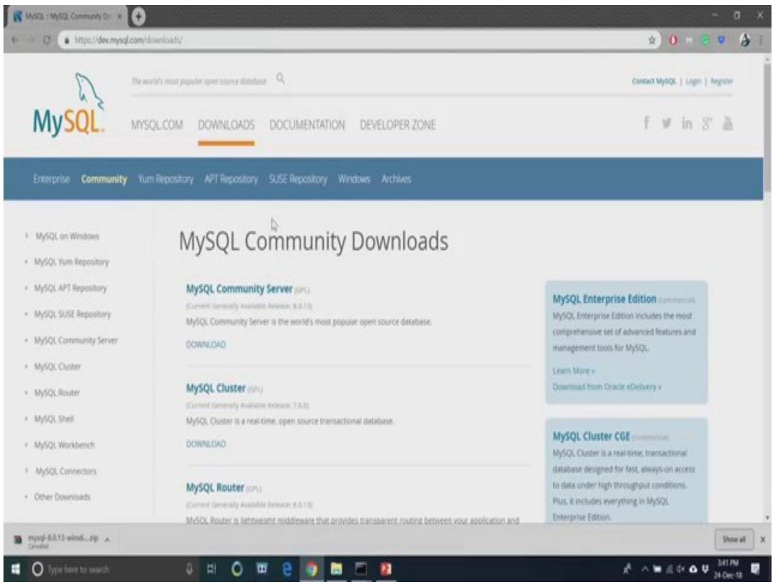
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
DEBASIS SAMANTA
CSE
IIT KHARAGPUR

So, JDBC is the middle which will basically connect your program means Java application to server. Now in this session, we will try to learn about the details about the JDBC driver.

So, first, we should see exactly for this MySQL server, the proper JDBC driver needs to be installed. So, we will discuss about how the JDBC driver will be responsible for connecting our Java application to the MySQL server. So, downloading the JDBC driver and then once the JDBC driver is installed successfully after downloading; obviously, the installation we will see exactly how this can be installed. And then finally, the through the JDBC driver and then Java program how we can execute some statement, SQL statement; and we will execute the SQL statement from our Java application Java program, and then we will see their results from the console again; that means, through the again connecting them is MySQL server after executing. This basically gives an idea about that whatever the comment that we are intended to execute form the application whether it is successfully executed or not.

In our next session that will be held later on we will see exactly how the results also that can be obtained purely from the application itself ok. Now let us start about how the JDBC driver can be installed. The location of the JDBC driver is we have given the link is there.

(Refer Slide Time: 02:11)



MySQL Community Downloads

MySQL Community Server (GPL)
(Current Generally Available Release: 8.0.13)
MySQL Community Server is the world's most popular open source database.

MySQL Cluster (GPL)
(Current Generally Available Release: 7.6.8)
MySQL Cluster is a real-time, open source transactional database.

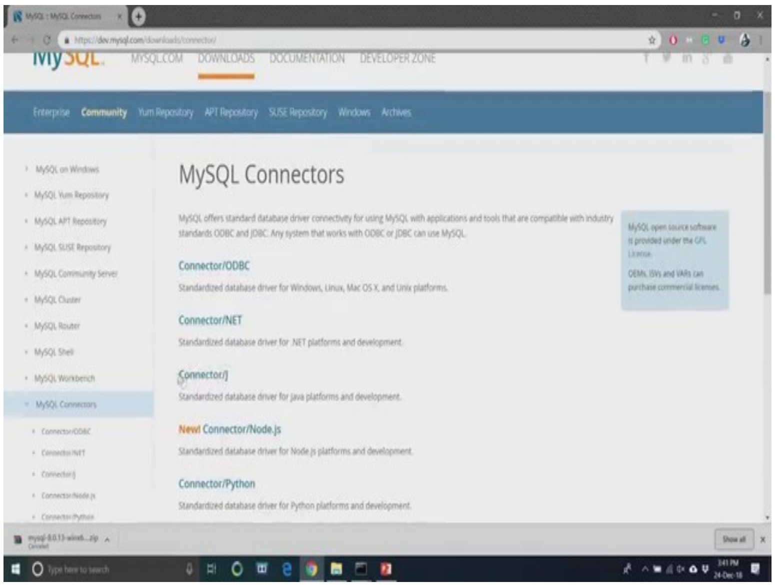
MySQL Router (GPL)
(Current Generally Available Release: 8.0.13)
MySQL Router is lightweight middleware that provides transparent routing between your application and

MySQL Enterprise Edition (commercial)
MySQL Enterprise Edition includes the most comprehensive set of advanced features and management tools for MySQL.

MySQL Cluster CGE (commercial)
MySQL Cluster is a real-time, transactional database designed for fast, always-on access to data under high throughput conditions. Plus, it includes everything in MySQL Enterprise Edition.

So, https://dev.mysql.com/downloads/connectors/j/. So, this basically is the link from where you can get the JDBC driver for the MySQL connector.

(Refer Slide Time: 02:29)



MySQL Connectors

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. Any system that works with ODBC or JDBC can use MySQL.

Connector/ODBC
Standardized database driver for Windows, Linux, Mac OS X, and Unix platforms.

Connector/NET
Standardized database driver for .NET platforms and development.

Connector/J
Standardized database driver for Java platforms and development.

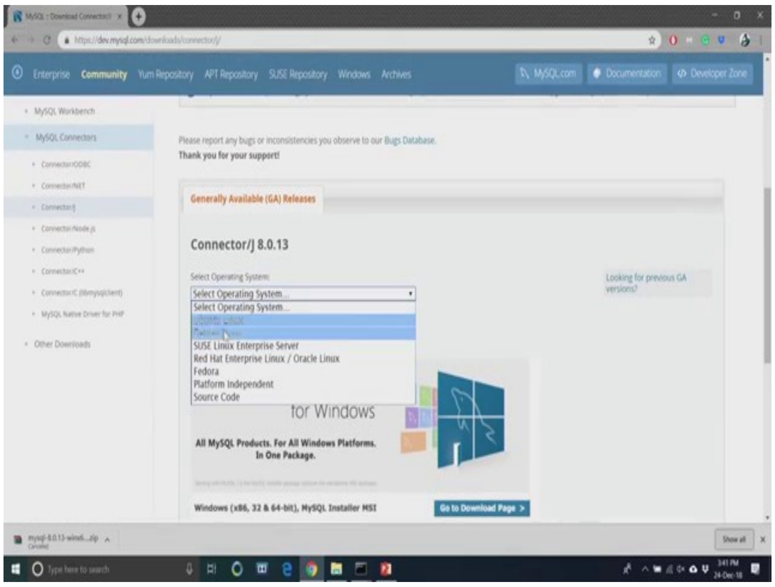
New! Connector/Node.js
Standardized database driver for Node.js platforms and development.

Connector/Python
Standardized database driver for Python platforms and development.

MySQL open source software is provided under the GPL License.
OEMs, ISVs and VARs can purchase commercial licenses.

So, MySQL connector is there and there is a connector J.

(Refer Slide Time: 02:35)



Please report any bugs or inconsistencies you observe to our Bugs Database.
Thank you for your support!

Generally Available (GA) Releases

Connector/J 8.0.13

Select Operating System:
Select Operating System...
Select Operating System...
SUSE Linux Enterprise Server
Red Hat Enterprise Linux / Oracle Linux
Fedora
Platform Independent
Source Code

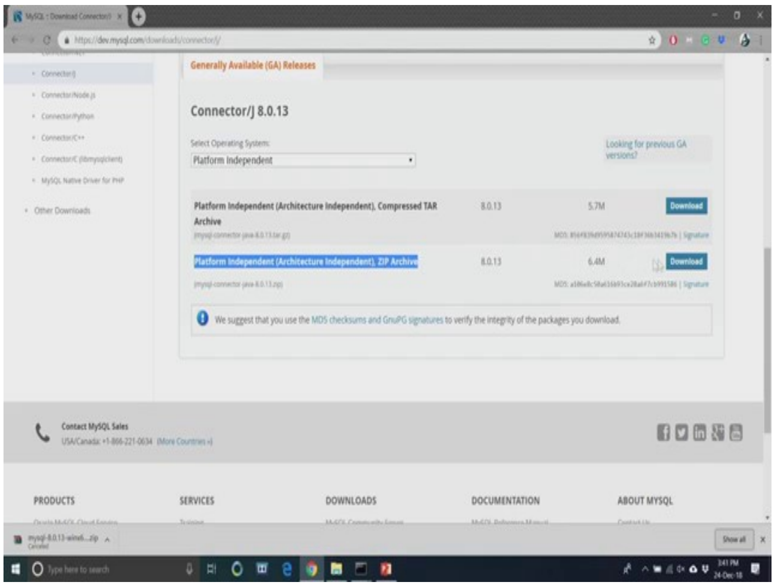
Looking for previous GA versions?

for Windows
All MySQL Products. For All Windows Platforms. In One Package.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

You just go to the link and finally, you will be able to set the proper program there.

(Refer Slide Time: 02:41)



Generally Available (GA) Releases

Connector/J 8.0.13

Select Operating System: Platform Independent

Looking for previous GA versions?

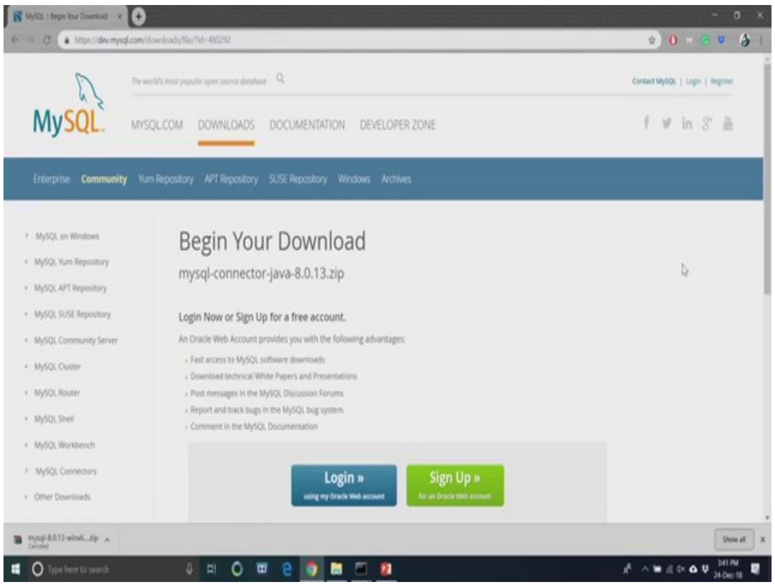
Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.13.tar.gz)	8.0.13	5.7M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.13.zip)	8.0.13	6.4M	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Contact MySQL Sales
USA/Canada: +1-866-221-0634 (More Countries »)

So, it is basically a platform-independent one version that you have to install it. And so, this is the version this is a zip file of course, and the version is for current SQL 8.013.

(Refer Slide Time: 02:51)



Begin Your Download

mysql-connector-java-8.0.13.zip

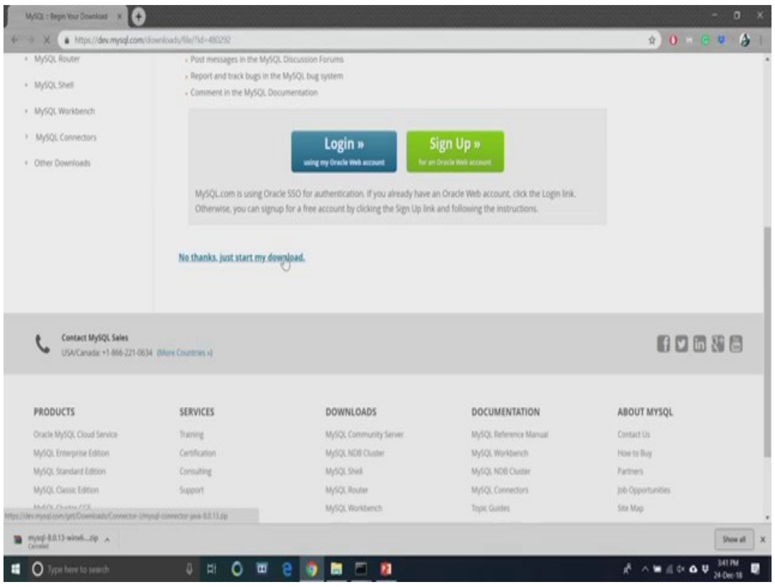
Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

It just simply starts downloading. So, you basically download now.

(Refer Slide Time: 02:53)


MySQL : Begin Your Download

https://dev.mysql.com/downloads/file/?id=480292

- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Connectors
- Other Downloads

- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

Login » using my Oracle Web account

Sign Up » for an Oracle Web account

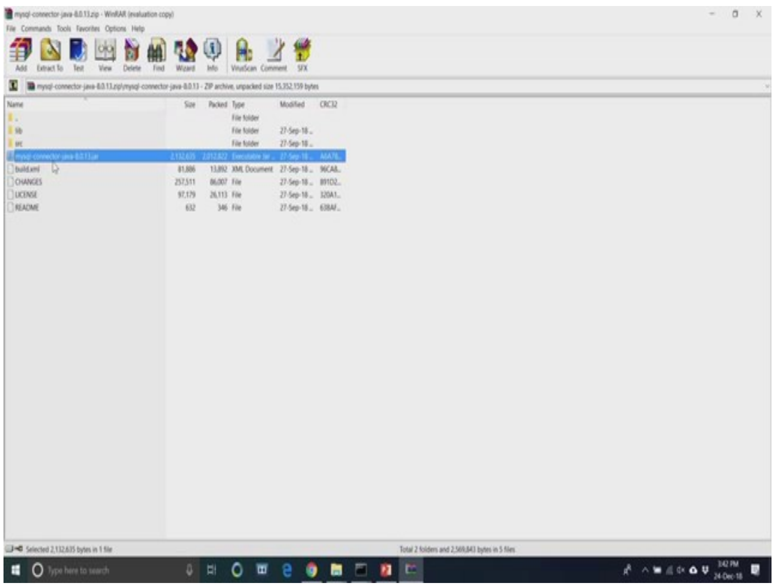
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

Contact MySQL Sales
USA/Canada: +1-866-221-0634 (More Countries »)

So, download from the net, it is free software. So, there is no issue. So, for downloading, this software is concerned. Once the JDBC driver is download then you have to extract the zip file, and it can we download in any directory. So, we will just advise you to download into on C drive right ok. So, we just unzip this zip file and then we, once you do the unzip, then you can find so many folders there are a one folders called JDBC MySQL connector.jar file.

(Refer Slide Time: 03:29)


mysql-connector-java-8.0.13.zip - WinRAR (evaluation copy)

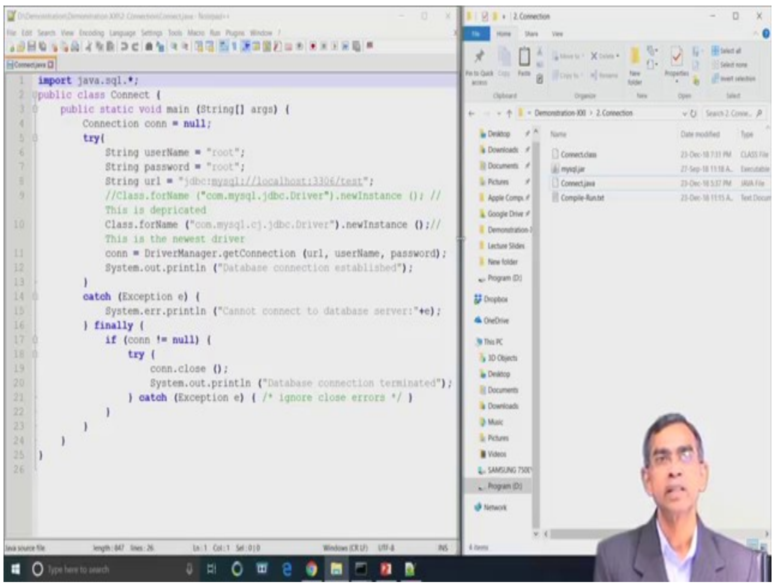
mysql-connector-java-8.0.13.zip\mysql-connector-java-8.0.13 - ZIP archive, unpacked size 15,352,159 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
lib			File folder	27-Sep-18 ...	
src			File folder	27-Sep-18 ...	
mysql-connector-java-8.0.13.jar	2,132,635	2,012,822	Executable Jar ...	27-Sep-18 ...	A6A76...
build.xml	81,886	13,892	XML Document	27-Sep-18 ...	96CA8...
CHANGES	257,511	86,007	File	27-Sep-18 ...	891D2...
LICENSE	97,179	26,113	File	27-Sep-18 ...	320A1...
README	632	346	File	27-Sep-18 ...	638AF...

So, this file you should rename to MySQL for just for simplicity, you can give any name, but for convenience you can give this one and then finally, you should place the jar file MySQL.jar file along with any other program which includes in this using JDBC, that should be done by one command from the command prompt java minus cp MySQL.jar and then what the all file name fine.

So, you just simply copy this jar file into this one right. So, we have copied it right. So, we have copied this MySQL jar file and this and then we have to finally, set the classpath name from the system file itself so that this JDBC driver can be identified from any location. If you execute it, so, you have to set the classpath also there.

(Refer Slide Time: 04:11)

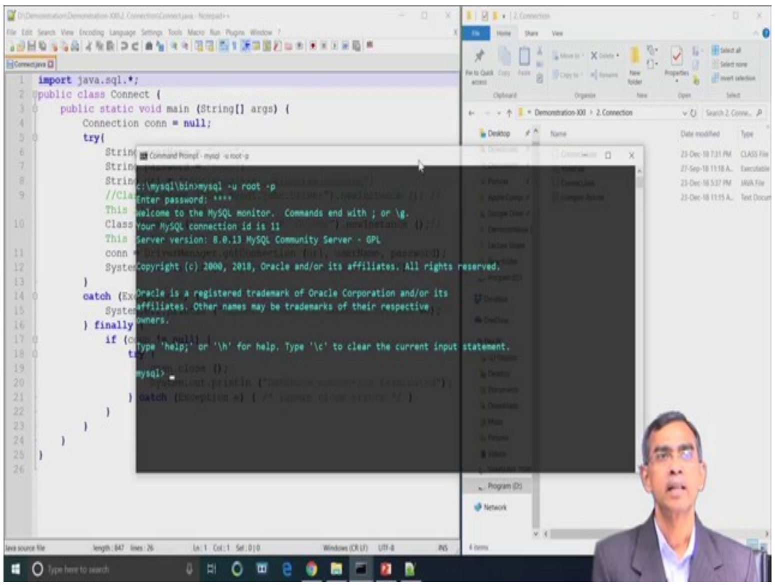


```java
import java.sql.*;
public class Connect {
    public static void main (String[] args) {
        Connection conn = null;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:mysql://localhost:3306/test";
            //Class.forName ("com.mysql.jdbc.Driver").newInstance (); //
            This is depricated
            Class.forName ("com.mysql.cj.jdbc.Driver").newInstance ();//
            This is the newest driver
            conn = DriverManager.getConnection (url, userName, password);
            System.out.println ("Database connection established");
        }
        catch (Exception e) {
            System.err.println ("Cannot connect to database server:"+e);
        } finally {
            if (conn != null) {
                try {
                    conn.close ();
                    System.out.println ("Database connection terminated");
                } catch (Exception e) { /* ignore close errors */ }
            }
        }
    }
}
```

Name	Date modified	Type
Connect.class	23-Dec-18 7:31 PM	CLASS File
mysql.jar	27-Sep-18 11:18 A...	Executable
Connect.java	23-Dec-18 5:37 PM	JAVA File
Compile-Run.txt	23-Dec-18 11:15 A...	Text Docum

Anyway, so, our JDBC connection this is this gives you complete competition about the JDBC connection is there.

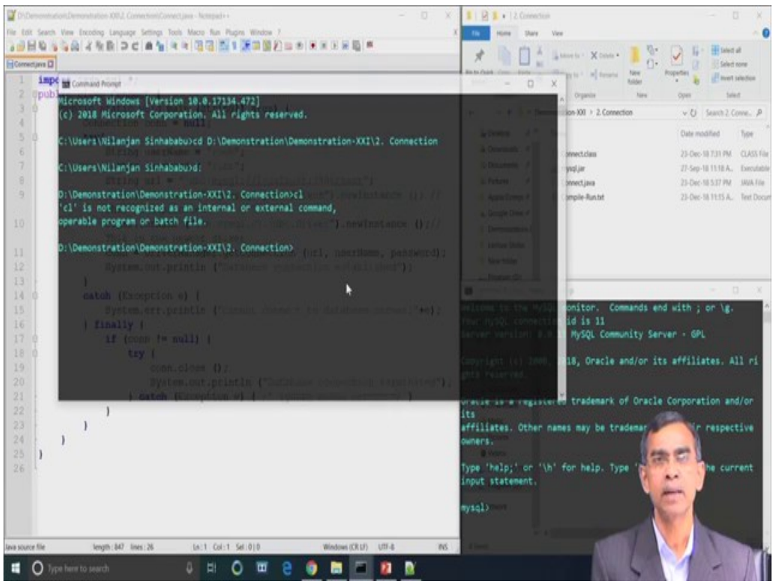
(Refer Slide Time: 04:37)



```
c:\mysql\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.13 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Now, you have to login to the MySQL server because we are quit from we are exited from this one, the password is root here now ok. So, the JDBC this MySQL connection is done currently it is there.

(Refer Slide Time: 04:51)

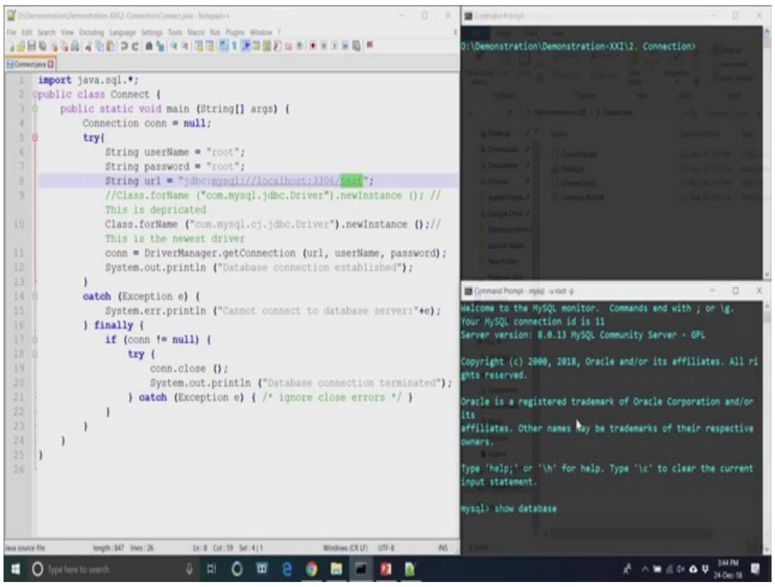


```
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nilanjan Sinhababu>cd D:\Demonstration\Demonstration-XXI\2. Connection

C:\Users\Nilanjan Sinhababu>d:

D:\Demonstration\Demonstration-XXI\2. Connection>cl
'cl' is not recognized as an internal or external command,
operable program or batch file.

D:\Demonstration\Demonstration-XXI\2. Connection>
```

Now, we want to write some Java program by which you can execute some SQL statements and then after the execution of the statement, we will be able to see the result. Now see there are few steps; obviously, involved. The first thing is that you have to create few objects; the first is that connection object we give the name as conn.

We define connection object and then we define ok. So, this is basically setting up the connection and then username is root, we have to write is username root password is root, URL is basically you can note it JDBC MySQL this is basically localhost and this is the number that is there, is a port number is 3306 and the JDBC, the drivers should connect to a particular database. So, the last field is basically the name of the database.

(Refer Slide Time: 05:49)



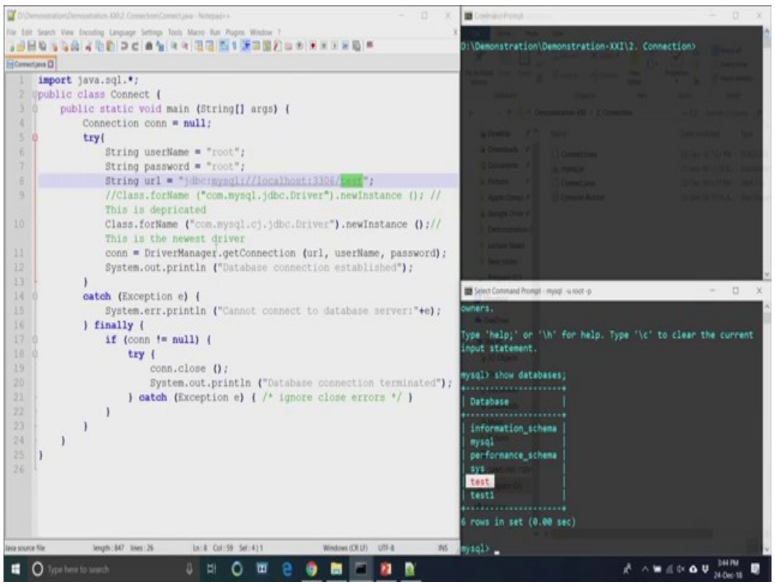
```java
import java.sql.*;
public class Connect {
    public static void main (String[] args) {
        Connection conn = null;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:mysql://localhost:3306/test";
            //Class.forName ("com.mysql.jdbc.Driver").newInstance (); //
            This is depricated
            Class.forName ("com.mysql.cj.jdbc.Driver").newInstance ();//
            This is the newest driver
            conn = DriverManager.getConnection (url, userName, password);
            System.out.println ("Database connection established");
        }
        catch (Exception e) {
            System.err.println ("Cannot connect to database server:"+e);
        } finally {
            if (conn != null) {
                try {
                    conn.close ();
                    System.out.println ("Database connection terminated");
                } catch (Exception e) { /* ignore close errors */ }
            }
        }
    }
}
```

```
D:\Demonstration\Demonstration-XXI\2. Connection>
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.13 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All ri
ghts reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> show database
```

As we have know already we have created one database in an earlier session, and the same database will be used there.

(Refer Slide Time: 05:59)



```java
import java.sql.*;
public class Connect {
    public static void main (String[] args) {
        Connection conn = null;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:mysql://localhost:3306/test";
            //Class.forName ("com.mysql.jdbc.Driver").newInstance (); //
            This is depricated
            Class.forName ("com.mysql.cj.jdbc.Driver").newInstance ();//
            This is the newest driver
            conn = DriverManager.getConnection (url, userName, password);
            System.out.println ("Database connection established");
        }
        catch (Exception e) {
            System.err.println ("Cannot connect to database server:"+e);
        } finally {
            if (conn != null) {
                try {
                    conn.close ();
                    System.out.println ("Database connection terminated");
                } catch (Exception e) { /* ignore close errors */ }
            }
        }
    }
}
```

```
D:\Demonstration\Demonstration-XXI\2. Connection>
```

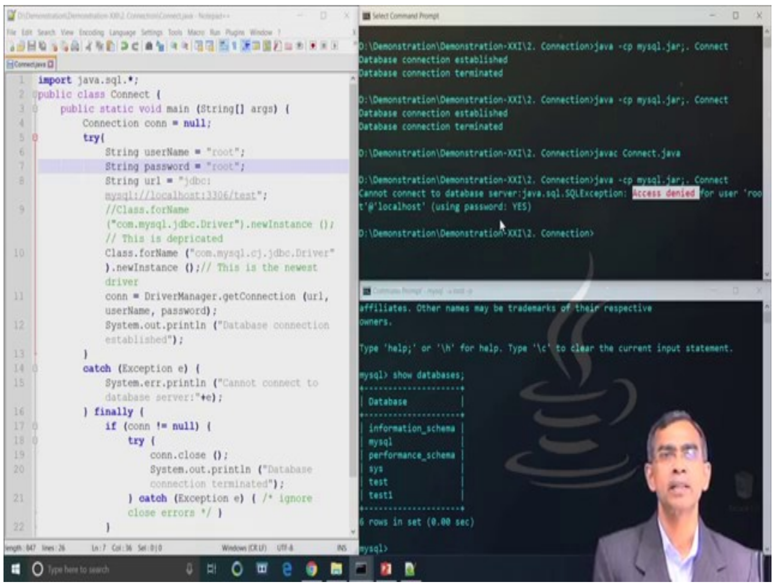
```
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
| test1              |
+--------------------+
6 rows in set (0.00 sec)

mysql> _
```

And then we have to load the JDBC driver by using the statement is basically the "Class.forName.com.MySQL.cj.jdbc.Driver".newInstance this is the method and so, this basically is to load the JDBC driver into your application actually ok. So, this is the right and then we have to establish the connection created an instance of the collection of object for which the DriverManager is the class there.

So, DriverManager.getConnection and giving the URL. Whatever the URL you have given there you just give the argument here URL, userName, password; otherwise you can in this method also you can type within double quote all the argument this also will work, but for the convenience, we store it and then pass it as a method arguments.

So, this basically creates the connection objects and then if the connection is successful from the program, we will get if there is no connection successful it will given an error. Now let us see whether error or not there. So, for the error handling, we have already enclose everything under the try-catch block. So, these are customary try-catch block that we have enclosed it now ok. So, run this program java minus ct yes.

(Refer Slide Time: 07:21)



```
import java.sql.*;
public class Connect {
    public static void main (String[] args) {
        Connection conn = null;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            //Class.forName
            ("com.mysql.jdbc.Driver").newInstance ();
            // This is depricated
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance ();// This is the newest
            driver
            conn = DriverManager.getConnection (url,
            userName, password);
            System.out.println ("Database connection
            established");
        }
        catch (Exception e) {
            System.err.println ("Cannot connect to
            database server:"+e);
        } finally {
            if (conn != null) {
                try {
                    conn.close ();
                    System.out.println ("Database
                    connection terminated");
                } catch (Exception e) { /* ignore
                close errors */ }
            }
```

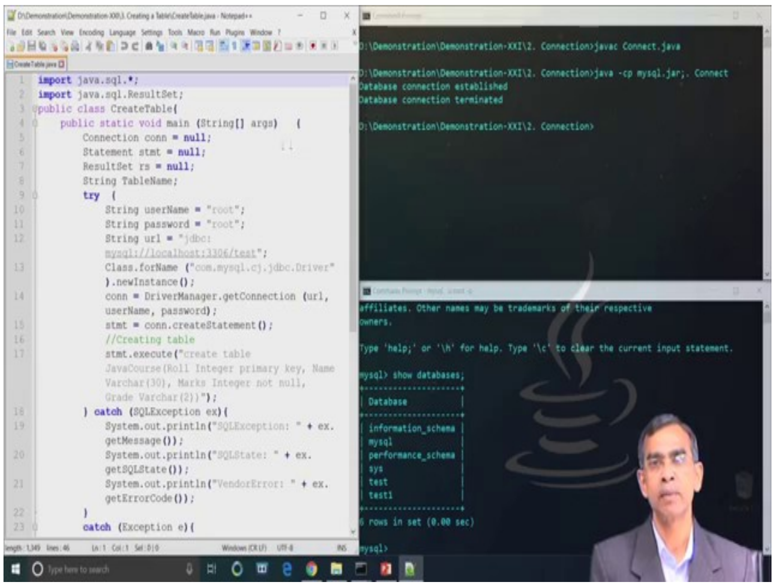
```
D:\Demonstration\Demonstration-XXI\2. Connection>java -cp mysql.jar;. Connect
Database connection established
Database connection terminated

D:\Demonstration\Demonstration-XXI\2. Connection>java -cp mysql.jar;. Connect
Database connection established
Database connection terminated

D:\Demonstration\Demonstration-XXI\2. Connection>javac Connect.java

D:\Demonstration\Demonstration-XXI\2. Connection>java -cp mysql.jar;. Connect
Cannot connect to database server:java.sql.SQLException: Access denied for user 'roo
t'@'localhost' (using password: YES)

D:\Demonstration\Demonstration-XXI\2. Connection>
```

```
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
| test1              |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

So, now we have already loaded this one here java minus cp MySQL.jar connect. Now connection is there. Now we can compile this program. So, the program is successfully compiled and executed after the execution is there. If we see the database connection established database connection is now terminated also.

Now, if I suppose there is a mistake for either say some name or some say root password is changed; so, guest g u e s t, suppose this is a wrong password then if wee again try to connect it, java minus cp MySQL.jar connect as you see it basically compile it. The compilation is done and then running this program, as you see cannot connect to database server this is because of this one. So, an exception is caused access denied for user root localhost using password yes. So, this basically refuses to get the connection. So, root is the password that is required for the correct or successful connection at this moment.

(Refer Slide Time: 08:39)



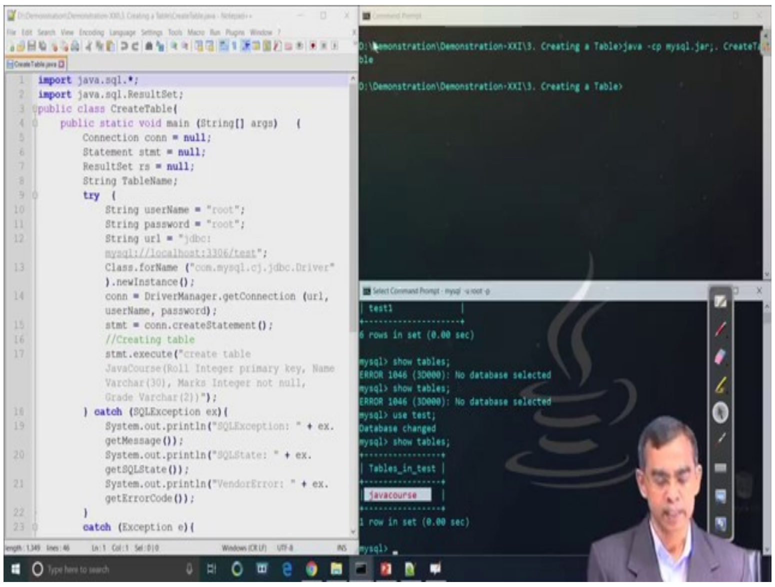
```java
import java.sql.*;
import java.sql.ResultSet;
public class CreateTable{
    public static void main (String[] args)   {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String TableName;
        try  {
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            //Creating table
            stmt.execute("create table
            JavaCourse(Roll Integer primary key, Name
            Varchar(30), Marks Integer not null,
            Grade Varchar(2))");
        } catch (SQLException ex){
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
        catch (Exception e){
```

```
D:\Demonstration\Demonstration-XXI\2. Connection>javac Connect.java

D:\Demonstration\Demonstration-XXI\2. Connection>java -cp mysql.jar;. Connect
Database connection established
Database connection terminated

D:\Demonstration\Demonstration-XXI\2. Connection>
```

```
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
| test1              |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

So, this way you can see how the connection can be made with the JDBC to the MySQL server through the JDBC there. Now we will discuss about how the simple statement can be executed. Here the simple statement that we are going to execute is, create a table java course because there is no table at the moment. So, as there is no table we will not be able to accomplish any insert, delete, update command.

(Refer Slide Time: 09:07)



```java
import java.sql.*;
import java.sql.ResultSet;
public class CreateTable{
    public static void main (String[] args)   {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String TableName;
        try  {
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            //Creating table
            stmt.execute("create table
            JavaCourse(Roll Integer primary key, Name
            Varchar(30), Marks Integer not null,
            Grade Varchar(2))");
        } catch (SQLException ex){
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
        catch (Exception e){
```

```
D:\Demonstration\Demonstration-XXI\3. Creating a Table>java -cp mysql.jar;. CreateTa
ble

D:\Demonstration\Demonstration-XXI\3. Creating a Table>
```

```
| test1              |
+--------------------+
6 rows in set (0.00 sec)

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> use test;
Database changed
mysql> show tables;
+----------------+
| Tables_in_test |
+----------------+
| javacourse     |
+----------------+
1 row in set (0.00 sec)

mysql> _
```

So, first, we have to create the table. Now, this is the one program as we see. This program basically tells how we can create a table. Now here you see this is the Connection object is created and then the Statement object is created of the class Statement, ResultSet it basically it will return result set; in this case, we will not consider all though, but it can give the results. And here is basically userName; root and then password also root we have stored and then URL is JDBC here; JDBC MySQL localhost this is basically URL for the our MySQL server in this case.

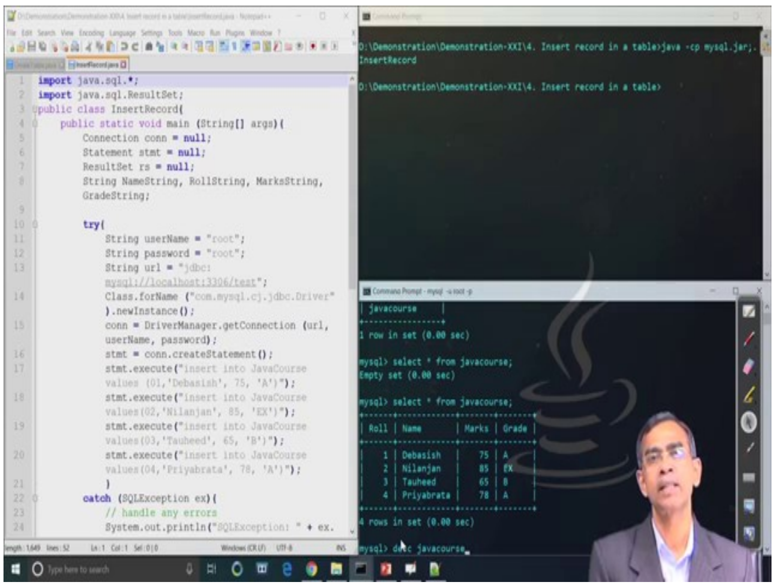
Then we establish the connection using this command Class.forName.com MySQL.cj.jdbc.Driver then .newInstance. This is basically the syntax for which the connection can be made. When the connection is it is successful then we have to create the connection object through the DriverManager getConnection giving the argument; URL, userName and password which is basically assigned here. And then so, this basically completes the connection and here is basically we are creating a statement object.

So, statement for conn.createStatement object is created and then this statement object is initialized with one statement the create table; JavaCourse Roll Integer primary key, Name Varchar 30, Marks Integer not null, Grade Varchar 2. So, it is basically this is the way the usual way, that a table can be created with four fields namely roll number name marks and grade like.

And so, now so, this basically completes the firing the execution and then if there is an error or whatever it is there it will obviously, catch by the try-catch block. So, this basically is the program which basically creates a connection objects connect to the server through JDBC driver and finally, execute a statement to create a new table and the command is that java minus cp MySQL.jar and then create a table this is the name of the class that we have created here.

So, this is basically the program and then in the server, we are coming to the console to the server and here show table-use test show table. So, we see the new table java course has been created under the test database. So, this is the first step or creating the table here. Now let us execute some other statements from this table. First is that we have to insert a few records into the table.

(Refer Slide Time: 11:53)



```java
import java.sql.*;
import java.sql.ResultSet;
public class InsertRecord{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;

        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("insert into JavaCourse
            values (01,'Debasish', 75, 'A')");
            stmt.execute("insert into JavaCourse
            values(02,'Nilanjan', 85, 'EX')");
            stmt.execute("insert into JavaCourse
            values(03,'Tauheed', 65, 'B')");
            stmt.execute("insert into JavaCourse
            values(04,'Priyabrata', 78, 'A')");
            }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
```

```
D:\Demonstration\Demonstration-XXI\4. Insert record in a table>java -cp mysql.jar;.
InsertRecord

D:\Demonstration\Demonstration-XXI\4. Insert record in a table>
```

```
| javacourse     |
+----------------+
1 row in set (0.00 sec)

mysql> select * from javacourse;
Empty set (0.00 sec)

mysql> select * from javacourse;
+------+------------+-------+-------+
| Roll | Name       | Marks | Grade |
+------+------------+-------+-------+
|    1 | Debasish   |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    3 | Tauheed    |    65 | B     |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql> desc javacourse
```

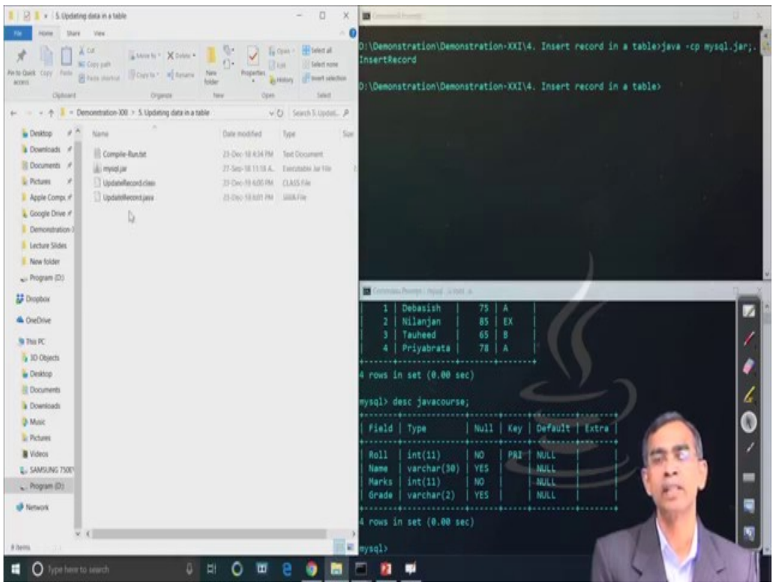
So, we will see exactly the table that you have just now created from our Java application how the same can be used to insert some data, we will create many records as many we wish. Here we will consider four records to be inserted with one SQL statement. Now, this is a program as you see this is a program. You see here first of all these are the. So, this is the conn is the Connection object, the STMT is the Statement object and then here is the basically userName and password and URL these are the three parameters that required to install the DriverManager.

So, we use these object and then finally, through DriverManager we connect it. So, the conn object is now ready; that means, its connection is established and then we just create a statement object and then the statement is basically created with execution for request with four insert records one by one. Here you see this is the first insert then semicolon, then the second statement.executes second insert and another statement.execute the third insert and then the fourth insert for this fourth execute statement.

So, this way we have issued the four a statement and then we call the executive method for this statement object to execute these are the four statement. This basically the request will pass through JDBC to the server, and the server will execute it and then finally, we will see whether a server has done the things correctly according to the request that is there in the application.

So, here again, we are running the program, this is the name of the classic insert records. So, right in ok. So, the common is java minus cp MySQL.jar that you have to give it colon then you have to give the InsertRecord. This is the syntax that you have to follow if you want to communicate. So, there is no error message that means it has been executed. So, let us come back to the table here again. So, we can see a select star from the java course, it is already there. Here also if we write say desc table java course also it will give the same things as usual because the table has been successfully created with the structure that we have given it there.

(Refer Slide Time: 14:21)



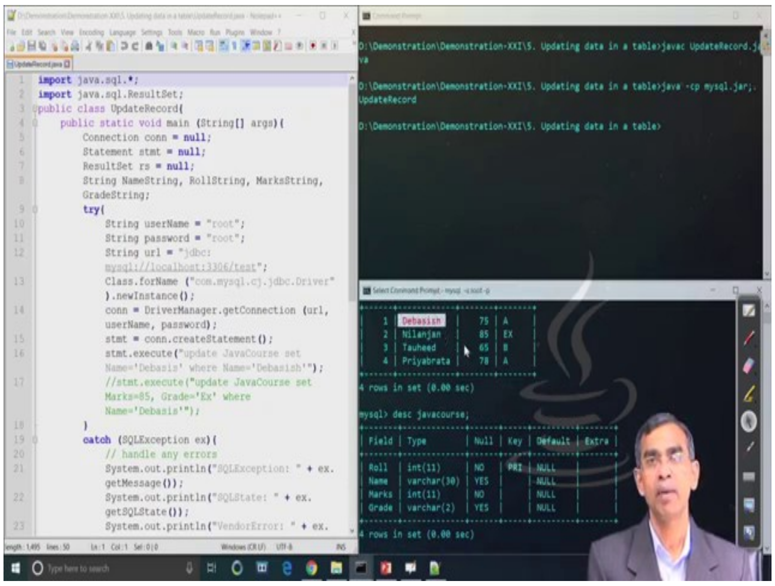
```
Compile-Run.txt
mysql.jar
UpdateRecord.class
UpdateRecord.java
```

```
D:\Demonstration\Demonstration-XXI\4. Insert record in a table>java -cp mysql.jar;.
InsertRecord

D:\Demonstration\Demonstration-XXI\4. Insert record in a table>
```

```
|    1 | Debasish   |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    3 | Tauheed    |    65 | B     |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql> desc javacourse;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Roll  | int(11)     | NO   | PRI | NULL    |       |
| Name  | varchar(30) | YES  |     | NULL    |       |
| Marks | int(11)     | NO   |     | NULL    |       |
| Grade | varchar(2)  | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

So, it is successfully created. Now so, the table is created four records has been have been inserted into it. Now we will see exactly the Update command how this can be executed from the Java application through JDBC driver to MySQL.

(Refer Slide Time: 14:43)



```java
import java.sql.*;
import java.sql.ResultSet;
public class UpdateRecord{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("update JavaCourse set
            Name='Debasis' where Name='Debasish'");
            //stmt.execute("update JavaCourse set
            Marks=85, Grade='Ex' where
            Name='Debasis'");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
```

```
D:\Demonstration\Demonstration-XXI\5. Updating data in a table>javac UpdateRecord.ja
va

D:\Demonstration\Demonstration-XXI\5. Updating data in a table>java -cp mysql.jar;.
UpdateRecord

D:\Demonstration\Demonstration-XXI\5. Updating data in a table>
```

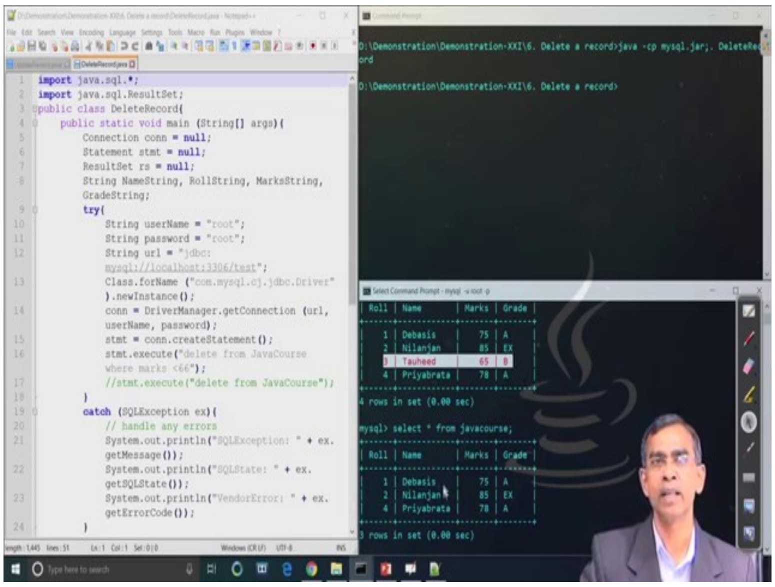
```
+------+------------+-------+-------+
|    1 | Debasish   |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    3 | Tauheed    |    65 | B     |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql> desc javacourse;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Roll  | int(11)     | NO   | PRI | NULL    |       |
| Name  | varchar(30) | YES  |     | NULL    |       |
| Marks | int(11)     | NO   |     | NULL    |       |
| Grade | varchar(2)  | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Again let us look at the code here ok. This code is, as usual, this is the Connection, Statement objects and these are userName, password and then this is the URL and we create the Class.forName passing the userName, password and then URL.

And then finally, we create the conn object; that means, connection object in calling the DriverManager.getConnection method with the URL, userName and password and then this is the statement object is now ready to fire. Now here the statement you see we use which command, SQL command to include it here update JavaCourse set. Name equals to this one where Name equals to this one. So, here is basic and you see in the first table that we have discussed there is one entry basically the first entry for which the name is Debasish d e b a s i s h. So, it is basically this command is requested to change this entry as a d e b a s i s. So, now, we are this program that is now ready and then let us compile this program and then ok.

So, this program yes. So, update record this is the name of the file yes. Now the record has been updated and let us come back to the command prompt. So, command prompt again just see exactly select star from you can see the updation is successful. So, this way the command relative update can be fired from the Java application and then we will see exactly delete command, we will just be going to delete one record from the table is java course here and here.

(Refer Slide Time: 16:37)



```java
import java.sql.*;
import java.sql.ResultSet;
public class DeleteRecord{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("delete from JavaCourse
            where marks <66");
            //stmt.execute("delete from JavaCourse");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
```

```
D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>
```

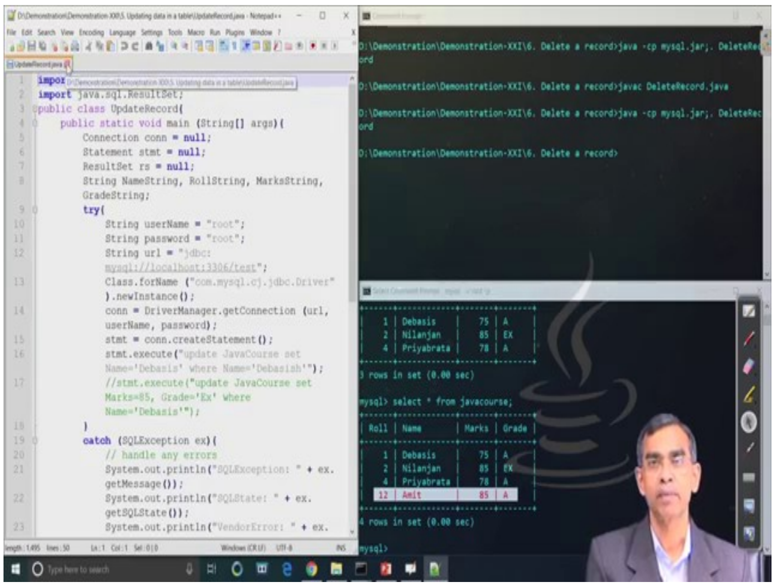
```
| Roll | Name       | Marks | Grade |
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    3 | Tauheed    |    65 | B     |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql> select * from javacourse;
+------+------------+-------+-------+
| Roll | Name       | Marks | Grade |
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
3 rows in set (0.00 sec)
```

So, it is basically this is the program as we see. So, this is the usual the things it is there in we have already familiar with this thing in the last examples and this is also clear only I just changed that we have done here. So, we have just create delete from JavaCourse where marks are less than 66.

So, in the record, there are some marks if it is less than 66 like the which 65 it will be deleted. Now so, let us. So, this is the only statement that we are going to do it. So, delete records. So, it is there now as you see there. Now here I am just going to change one little bit small change here. So, as you have already seen that. So, more than one SQL statement can be executed. So, those actually this is basically a statement.execute method is there.

So, I am just changing you just add one more statement.execute instead of delete keep it delete here, then insert right see roll number 7 Amit marks is not here, next just copy statement.execute again right. So, these are deleted we are giving going to execute. So, insert right.

(Refer Slide Time: 18:11)



```
impor
import java.sql.ResultSet;
public class UpdateRecord{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("update JavaCourse set
            Name='Debasis' where Name='Debasish'");
            //stmt.execute("update JavaCourse set
            Marks=85, Grade='Ex' where
            Name='Debasis'");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
```

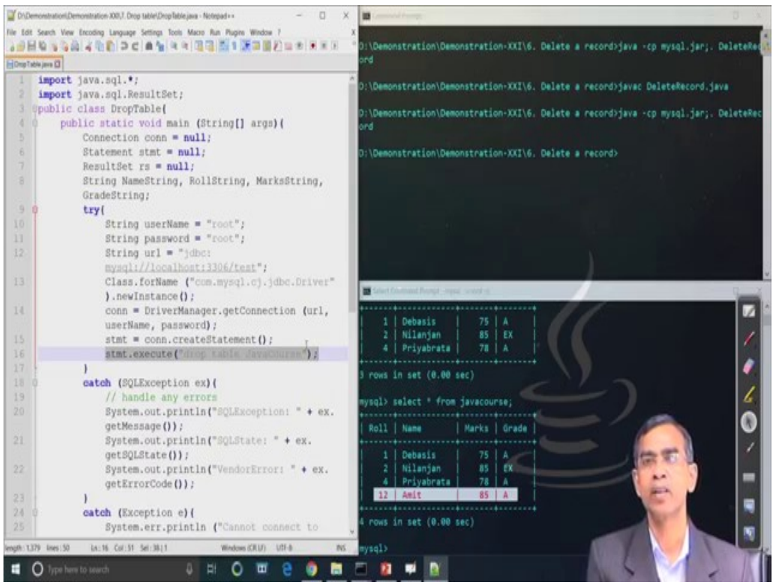
```
D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>javac DeleteRecord.java

D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>
```

```
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
3 rows in set (0.00 sec)

mysql> select * from javacourse;
+------+------------+-------+-------+
| Roll | Name       | Marks | Grade |
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    4 | Priyabrata |    78 | A     |
|   12 | Amit       |    85 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql>
```

Insert into the right java course then 0 name roll value yes. So, fine, so, Amit. So, values it should be valued inserting to java course values v a l u e s values 12 Amit, 85, a this one right. Now we are inserting one more record now I am just right going to here. So, we have to save this program we have changed it. Now it will execute two things; delete as well as insert also. So, yes nowhere you can see right. So, the new entry has been made here as you see. So, as many as SQL statement can be executed from one program, as you have seen you have tested it to two; so, it is not necessarily limited to two more than two statement also can be executed from the same program itself.

So, depending on your requirement whatever the SQL statements are there, that we will be able to insert it.

(Refer Slide Time: 20:05)



```java
import java.sql.*;
import java.sql.ResultSet;
public class DropTable{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("drop table JavaCourse");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
        catch (Exception e){
            System.err.println ("Cannot connect to
```

```
D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>javac DeleteRecord.java

D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>
```

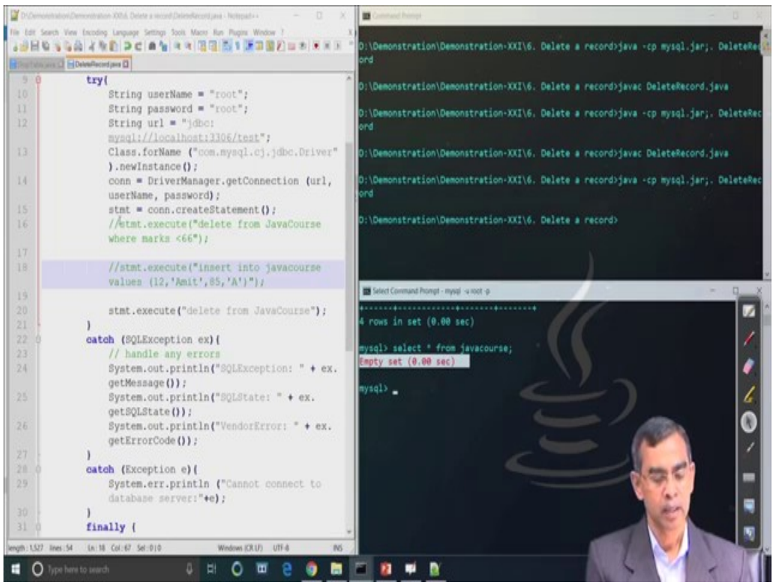
```
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    4 | Priyabrata |    78 | A     |
+------+------------+-------+-------+
3 rows in set (0.00 sec)

mysql> select * from javacourse;
+------+------------+-------+-------+
| Roll | Name       | Marks | Grade |
+------+------------+-------+-------+
|    1 | Debasis    |    75 | A     |
|    2 | Nilanjan   |    85 | EX    |
|    4 | Priyabrata |    78 | A     |
|   12 | Amit       |    85 | A     |
+------+------------+-------+-------+
4 rows in set (0.00 sec)

mysql>
```

Now here is the another delete statement as you know, we can delete the entire what is called the all records from the course also or so, if you want to delete all records, so, what is the command then? Delete all records.

Student: Drop tables (Refer Time: 20:19).

No, the drop table will delete anyway. So, now, there is another delete command; delete if you give the name of the database right. So, delete from JavaCourse.

(Refer Slide Time: 20:33)



```java
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            //stmt.execute("delete from JavaCourse
            where marks <66");

            //stmt.execute("insert into javacourse
            values (12,'Amit',85,'A')");

            stmt.execute("delete from JavaCourse");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
        catch (Exception e){
            System.err.println ("Cannot connect to
            database server:"+e);
        }
        finally {
```

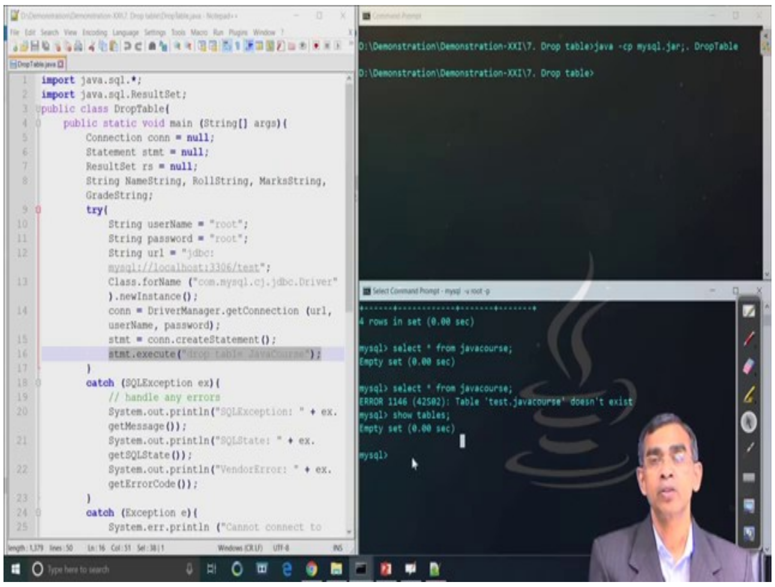
```
D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>javac DeleteRecord.java

D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>javac DeleteRecord.java

D:\Demonstration\Demonstration-XXI\6. Delete a record>java -cp mysql.jar;. DeleteRec
ord

D:\Demonstration\Demonstration-XXI\6. Delete a record>
```

```
+-------+------------+-------+--------+
4 rows in set (0.00 sec)

mysql> select * from javacourse;
Empty set (0.00 sec)

mysql>
```

So, this means that it will delete all the records from the JavaCourse. So, it is the command it is there, we are changing like delete from now delete from JavaCourse as this is the statement that we have executed from this one. So, it will basically delete all the records. So, now, the record is empty.

So, there are no records. If you again insert as you see another record will be inserted or whatever it is there, and if it is select star also it will not return anything because it is it does not have any record there.

(Refer Slide Time: 21:17)



```java
import java.sql.*;
import java.sql.ResultSet;
public class DropTable{
    public static void main (String[] args){
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        String NameString, RollString, MarksString,
        GradeString;
        try{
            String userName = "root";
            String password = "root";
            String url = "jdbc:
            mysql://localhost:3306/test";
            Class.forName ("com.mysql.cj.jdbc.Driver"
            ).newInstance();
            conn = DriverManager.getConnection (url,
            userName, password);
            stmt = conn.createStatement();
            stmt.execute("drop table JavaCourse");
        }
        catch (SQLException ex){
            // handle any errors
            System.out.println("SQLException: " + ex.
            getMessage());
            System.out.println("SQLState: " + ex.
            getSQLState());
            System.out.println("VendorError: " + ex.
            getErrorCode());
        }
        catch (Exception e){
            System.err.println ("Cannot connect to
```

```
D:\Demonstration\Demonstration-XXI\7. Drop table>java -cp mysql.jar;. DropTable

D:\Demonstration\Demonstration-XXI\7. Drop table>
```

```
+-------+------------+-------+--------+
4 rows in set (0.00 sec)

mysql> select * from javacourse;
Empty set (0.00 sec)

mysql> select * from javacourse;
ERROR 1146 (42S02): Table 'test.javacourse' doesn't exist
mysql> show tables;
Empty set (0.00 sec)

mysql>
```

So, finally, we can delete the entire table also drop table and then. So, this is the program here drop table here in this program. So, a statement can be executed and then if we show tables thereafter the drop table program is executed successfully.

So, the drop table will permanently delete the entire table actually and you see there is no table here. So, the selects and show tables also if you see, so, it does not have any tables in this way. Now so, what we have learned here is that how we can establish a connection and then after the connection is successful then we will be able to execute the SQL statement from Java application.

So, in this demonstration although we have to use the localhost, but the URL as you know the URL specification that we have discussed during our networking discussion, the complete specification URL using the protocol and then resource name and everything if we mention properly including port number etcetera, then it will basically redirect your aggregation to that particular remote server whether it is connected through the network or it is the localhost absolutely this is the no issue you can use the same way actually.

So, in this case, all through remote connection is not possible to establish at the moment although you could do that, but anyway you have done it, but the procedure is same only the URL needs to be changed. And you should have the privilege to access your SQL server using your username and password that also should be ready with you, then only you will be able to establish any connection and thereby execute any SQL statement.

So, this is for today's plan and in our next session, we will discuss about how all the results that basically after executing a particular SQL statement, which will return the SQL server will be handled by our Java application itself. So, that will be discussed in our next session.

Thank you very much.