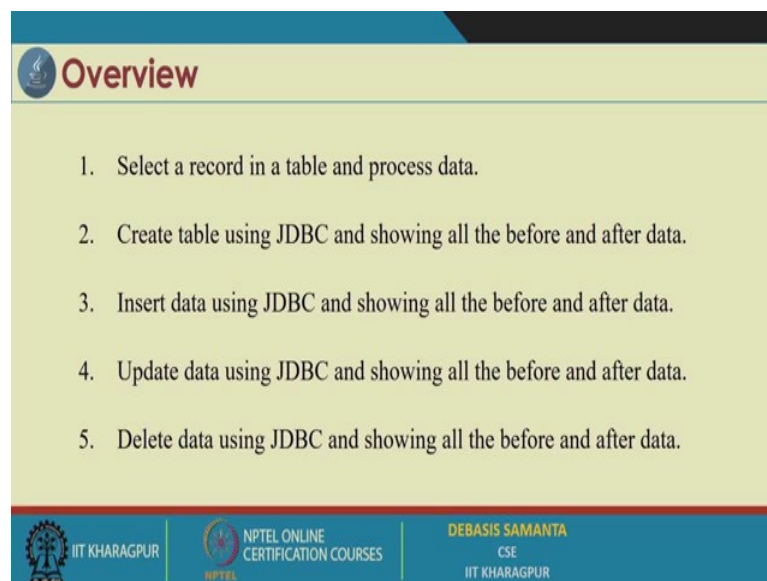


Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 55
Demonstration – XXII

I hope you are enjoying our Demonstrations. This is our last demo and this demo is related to the lessons that we have learn so far the JDBC concept is concerned.

(Refer Slide Time: 00:31)



Overview

1. Select a record in a table and process data.
2. Create table using JDBC and showing all the before and after data.
3. Insert data using JDBC and showing all the before and after data.
4. Update data using JDBC and showing all the before and after data.
5. Delete data using JDBC and showing all the before and after data.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHARAGPUR

Now, to in today's demo we basically going to cover about that not exactly connecting to the SQL server on console rather program to program only. So, that from the java application itself how can issue different SQL commands and the commands can be sent to the server, server will execute the command and then the server will return the results and all the results will be processed from the application itself. So, everything is basically included in the application only we will not go out from the application.

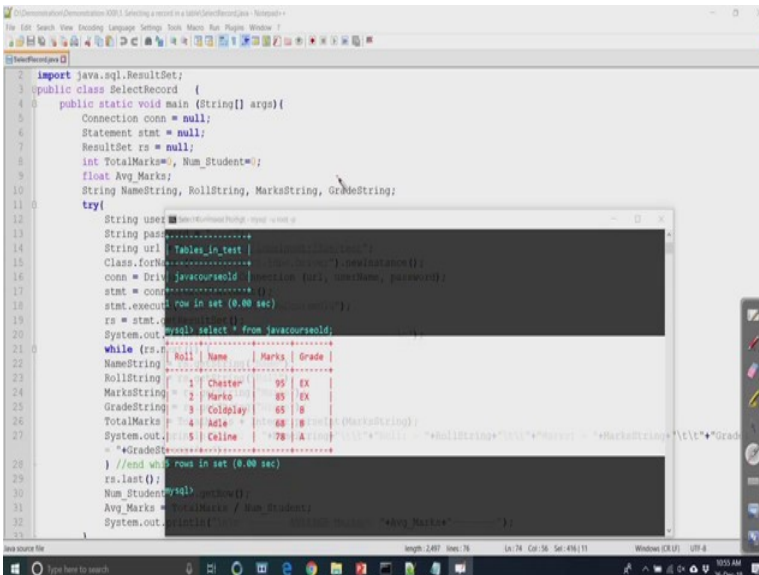
Now, so far this thing is concerned, we will see exactly how the different statements can be executed and then they can be fire to the server. And you know exactly as a process, their first step that you have to consider is that you have to first create three objects. One is the Connection objects, the Statement object and then Resultset objects. So, we have already learnt about how the connection objects can be created and then from your application how the connection can be established to a server through JDBC driver.

And then we have also learned about in our last session that how the JDBC driver can be used to have the communication that mean from application to the server some request can be issued. Today we will discuss, now we are going to discuss about the program; the statement exhibition of the different SQL statements and then getting the results and processing the results in the application environment itself.

Now, we are going to discuss about how the execution of a select statement can be done and after the select statement as you know, it will return a set of records, how all those records can be processed from the application itself. For example, display on the screen only. So, far the processing is concerned we will consider simple processing. In the first example we will consider the data will be faced from the server and the data some calculation for example, average calculation of all the records will be calculated like this one. And then we will discuss about how the table can be created and then after the creation of successful table from the program itself, we will be able to confirm our self that the table has been created successfully.

And then we will see exactly how the insert command can be executed and after the completion of the insert command execution we can ensure our self that insert of data is successfully achieved and then finally, with the repeat for the same other commands like update, delete, drop table etcetera.

(Refer Slide Time: 03:23)



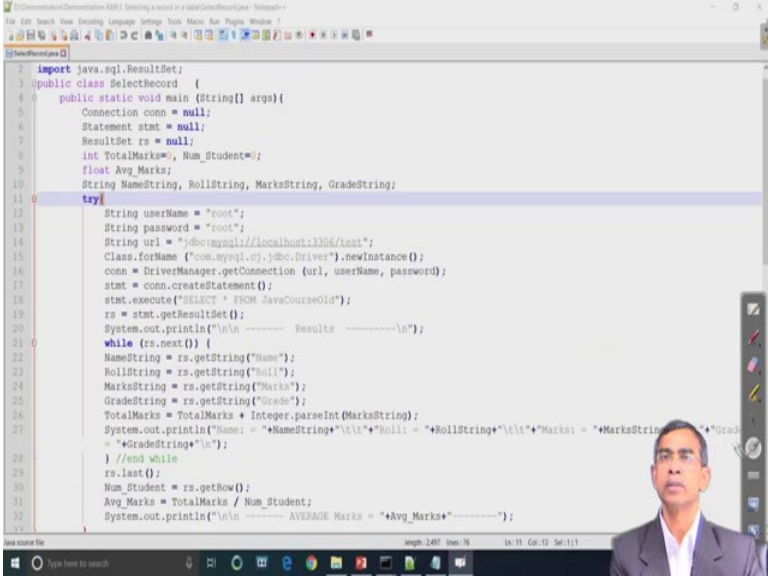
```
1 import java.sql.*;
2
3 public class SelectRecord {
4     public static void main (String[] args){
5         Connection conn = null;
6         Statement stat = null;
7         ResultSet rs = null;
8         int TotalMarks=0, Num_Student=0;
9         float Avg_Marks;
10        String NameString, RollString, MarksString, GradeString;
11        try{
12            String user = "root";
13            String pass = "root";
14            String url = "jdbc:mysql://localhost:3306/javacourseold";
15            Class.forName("com.mysql.jdbc.Driver");
16            conn = DriverManager.getConnection(url, user, pass);
17            stat = conn.createStatement();
18            stat.executeUpdate("row in set (0.00 sec)");
19            rs = stat.executeQuery("select * from javacourseold");
20            System.out.println("select * from javacourseold");
21            while (rs.next()){
22                NameString = rs.getString(1);
23                RollString = rs.getString(2);
24                MarksString = rs.getString(3);
25                GradeString = rs.getString(4);
26                TotalMarks = TotalMarks + Integer.parseInt(MarksString);
27                System.out.println("Roll : " + RollString + " Name : " + NameString + " Marks : " + MarksString + " Grade : " + GradeString);
28            } //end while row in set (0.00 sec)
29            rs.close();
30            Num_Student=rs.getInt(1);
31            Avg_Marks = TotalMarks / Num_Student;
32            System.out.println("Avg_Marks : " + Avg_Marks);
33        }
34    }
35 }
```

So, let us proceed for the demonstration. The first program that we are going to demo is basically how the select statement can be executed. For these things we assume that there is already a table available in the database. We have already created a table for the purpose of this demo. The Name of the table is javacourseold. Now as a yeah, so, this is the table that it is there already stored in the MySQL server and we just applied operation select statement on this.

For example, select star from javacourseold. So, basically it will execute this statement and then on the successful execution it will produce a result, those result will be faced to the application and from the application, we will be able to process it.

Now, here is the program. Let us see. As you see the program here so, few statements as you know which are obvious like how to establish the connection.

(Refer Slide Time: 04:25)



```
1 import java.sql.ResultSet;
2 public class SelectRecord {
3     public static void main (String[] args){
4         Connection conn = null;
5         Statement stmt = null;
6         ResultSet rs = null;
7         int TotalMarks=0;
8         float Avg_Marks;
9         String NameString, RollString, MarksString, GradeString;
10        try{
11            String userName = "root";
12            String password = "root";
13            String url = "jdbc:mysql://localhost:3306/test";
14            Class.forName ("com.mysql.cj.jdbc.Driver").newInstance();
15            conn = DriverManager.getConnection (url, userName, password);
16            stmt = conn.createStatement();
17            stmt.execute("SELECT * FROM JavaCourseOld");
18            rs = stmt.getResultSet();
19            System.out.println("\n\n Results ----->\n");
20            while (rs.next()) {
21                NameString = rs.getString("Name");
22                RollString = rs.getString("Roll");
23                MarksString = rs.getString("Marks");
24                GradeString = rs.getString("Grade");
25                TotalMarks = TotalMarks + Integer.parseInt(MarksString);
26                System.out.println("Name = "+NameString+"\t\tRoll = "+RollString+"\t\tMarks = "+MarksString+"\t\tGrade = "+GradeString);
27            } //end while
28            rs.last();
29            Num_Student = rs.getRow();
30            Avg_Marks = TotalMarks / Num_Student;
31            System.out.println("\n\n ----- AVERAGE Marks = "+Avg_Marks+"-----");
32        }
33    }
34 }
```

So, as we see this is basically regarding the connecting to the server. So, as we see these are the connection is there and then finally, in this statement we are executing the statement. As you see the statement that we are going to execute select star from javacourseold. So, from this table it will be there. Now if this command is executed then this will basically produce a result. So, getResultSet will receive the result of the statement object and then it will be stored in another object call the rs. So, rs is a result set object is there. So, mainly three objects as I have already told you involved; connection object, statement object and result sets. We you have already familiar in our

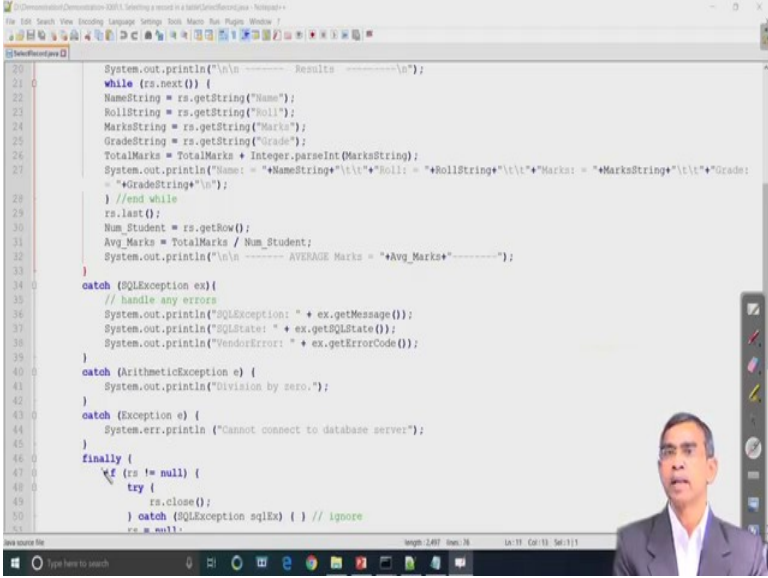
last two demonstration, we have discussed about it. Now we are going to process this rs objects.

Now, you know while rs dot next is basically written a Boolean value. So, it will basically rs objects contents a set of records that will basically receive. So, all the records it is there; that means, all rows we can say. Now rs dot next is basically till the end of the rows it will process. So, initially the rows in number of rows is not empty actually. So, it will the first row. So, it rs dot next is true actually. So, as it is true it will go to this one. So, rs dot getString is basically for the record in the javacourseold; there is a one field call Name then another is called Roll and Marks and Grade. So, for the current record the field value which has the field name Name will be obtained and it basically read as a string. So, java always read initially as a string and then we can convert into the other form that is per as per our requirement. So, is basically read the string for the field Name the value and we store is at a string.

So, similarly NameString, RollString, MarkString, GradeStrings are the four different field values for a record; namely, Name, Roll, Marks and Grade and then we then we convert all the values. For example, we are our objective is that once you read the marks we have to calculate the TotalMarks sum of all the marks for all the records and then finally, divide by the total number of records and then average will be calculated. So, this basically is process is basically TotalMarks initially 0 and then Integer dot parseInt MarkString because it is initially as a string and use this conversation we can convert the integer value and then TotalMarks will be added. So, these basically reach a record each time and then read the I mean at the values marks value to the TotalMarks and finally, the total of all marks will be obtained here and here actually next is that.

So, System dot out dot println Name equals NameString Roll RollString and then Marks is basically printed here. So, MarkString is there. And then next is basically rs dot last means is basically when you reach to the last object, so, rs dot last is basically there. And then number of students is basically can be obtained from the rs object itself by calling the method getRow method. So, rs dot getRow. So, it basically says the total number of records that we have read it. So, TotalMarks divided by this total number of records it basically gives you the average calculation.

(Refer Slide Time: 08:03)



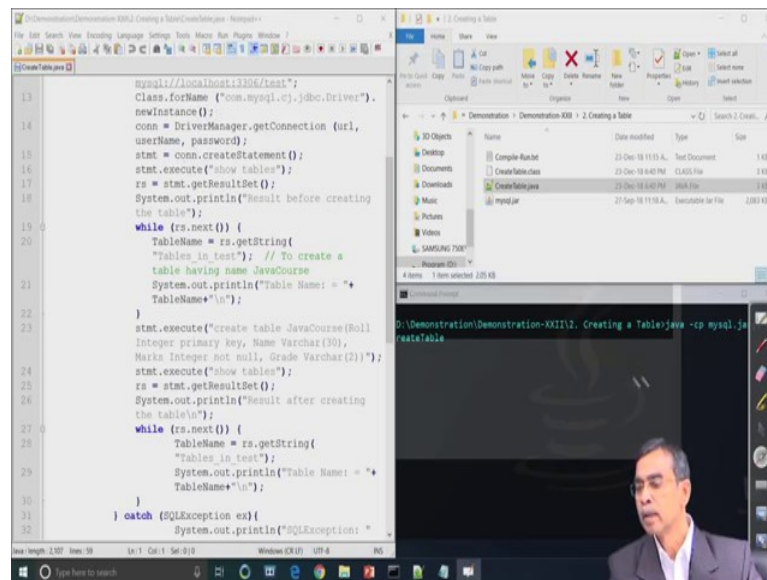
```
20 System.out.println("\n\n Results\n\n");
21 while (rs.next()) {
22     NameString = rs.getString("Name");
23     RollString = rs.getString("Roll");
24     MarksString = rs.getString("Marks");
25     GradeString = rs.getString("Grade");
26     TotalMarks = TotalMarks + Integer.parseInt(MarksString);
27     System.out.println("Name: " + NameString + "\t\t\tRoll: " + RollString + "\t\t\tMarks: " + MarksString + "\t\t\tGrade: " + GradeString + "\n");
28 } //end while
29 rs.last();
30 Num_Student = rs.getRow();
31 Avg_Marks = TotalMarks / Num_Student;
32 System.out.println("\n\n ----- AVERAGE Marks = " + Avg_Marks + "-----");
33
34 catch (SQLException ex){
35     // handle any errors
36     System.out.println("SQLException: " + ex.getMessage());
37     System.out.println("SQLState: " + ex.getSQLState());
38     System.out.println("VendorError: " + ex.getErrorCode());
39 }
40
41 catch (ArithmeticException e) {
42     System.out.println("Division by zero.");
43 }
44
45 catch (Exception e) {
46     System.err.println("Cannot connect to database server.");
47 }
48
49 finally {
50     if (rs != null) {
51         try {
52             rs.close();
53         } catch (SQLException sqlEx) { } // ignore
54     }
55 }
```

As we see the average calculation is the average is basically calculated here and then we print this average. And the rest of the thing is for the try, catch, block and everything this is the customary try, catch, block we have to maintain and so that the program can be executed in a reliable and robust manner.

So, these basically tell you that how the result set object can be processed. So, in this session we will mainly emphasize of the processing of the results such that the or alternatively the all result that will obtain after the execution of SQL statement from the application itself ok. So, this is the demonstration, hope you have you are able to understand it.

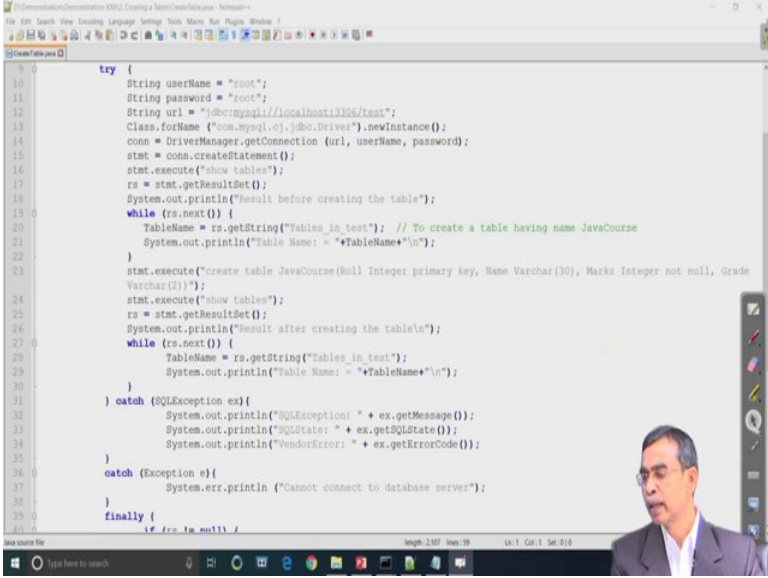
Our next demonstrations little bit simpler. We will see exactly how the different other SQL type of statement. We will consider only execution of basic SQL statements and this will be enough for the begin for the learning purpose only. So, in this case, we will consider one table already available.

(Refer Slide Time: 08:59)



Now, we will consider the create table right. So, we are going to consider how a new table can be created and then so, we first see first from the application itself, we will exactly check that what are the different tables are available there in the database server. So, in that case we have to issue the command. So, tables and then once the table is there we can create a new table. So, we just create a new table. The name of the new table maybe say java course and then after in creating a new table we will again see the confirmation about that new table has been created successfully. So, these are the steps that we will follow one by one, but from the java application itself ok. So, here is a program as we see go little bit ok.

(Refer Slide Time: 09:57)

A screenshot of a Java IDE window titled "Creating a Table/CreateTable.java - NetBeans". The code is as follows:

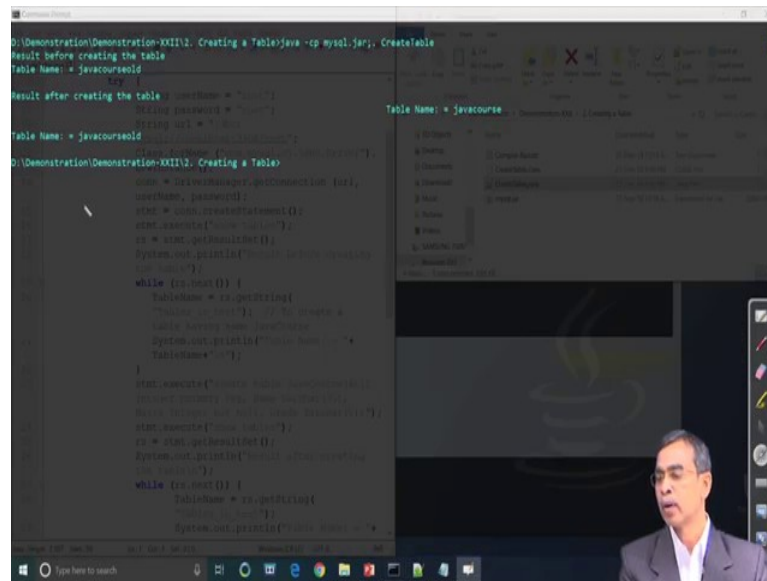
```
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

So, first few part of the program. So, first few part of the program is obvious is basically connecting to the server using JDBC. So, loading JDBC connecting all these things are there and then next is you see this is a statement that we are going to execute, show table command. Now so, the result set will be obtained. Basically show table will see you all the tables those are there in the current active database in this case may be test database. So, rs will basically get the result sets includes all the tables those are there in the database right now.

Now, the prompt is that results before creating the table and here is basically processing the rs objects is basically doing nothing is basically getting each TableName from the result sets and printing it on the screen that is all. So, it will basically do these things and then after the next thing we execute another statement create table JavaCourse.

And as you know this is the full syntax about how to create a table new table, in fact, and add into the database. So, is a statement execute and after this create a new table is created. The show table, it is basically send the previous code right after getting all the tables right; it will basically display all the tables. So, this is the part that ok, it will show table before the creating and then in create a new table and after the creation it will show again all the tables. Now, we will see the results that can be obtained from the java application. It is a as you see here from the java application we are able to see it.

(Refer Slide Time: 11:43)

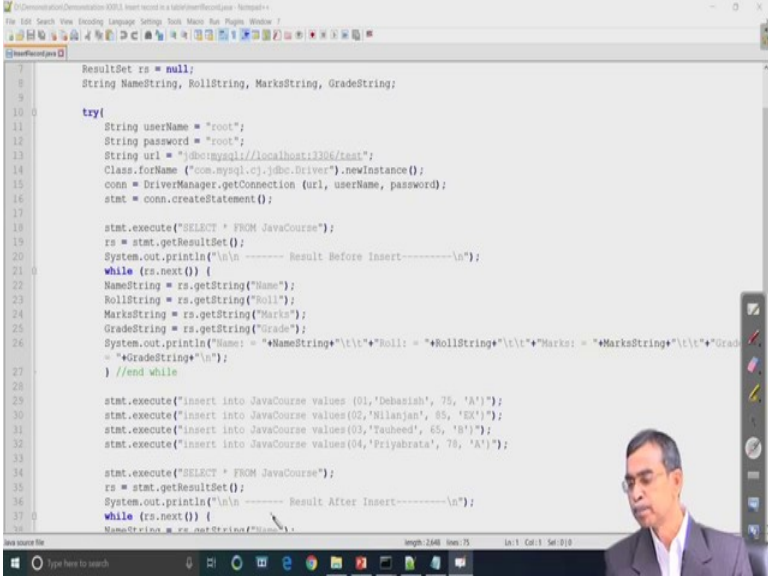


Now, if we run this program, the execution will flow look like this. So, this is the program and then ok. So, we just Table Name is a javacourse.

So these basically shows initially only one table is present there called the javacourseold. Now results after creating the table as you see after creating the table. So, we run this code again. After creating the table, it basically shows that javacourseold is there and then javacourse table is there. So, two tables are here; javacourseold the previous and then new one, it is there and then is basically is that basically this is the output actually as we see here. So, the statements have been executed. After the execution we can see the results and that results also through the JDBC itself.

Now, let us come to the next one another example where we are we can see exactly how we can insert a record into an already existing table. We will consider inserting record into the table say javacourseold. First we show before execution what is the content in the table and then we will again issue the insert statement and then after the insert statement the current data that is there.

(Refer Slide Time: 13:09)

A screenshot of a Java IDE window titled 'JDBCDemo\JDBCDemo\src\main\java\com\myorg\jdbc\JdbcMain.java'. The code is as follows:

```
7 ResultSet rs = null;
8 String NameString, RollString, MarksString, GradeString;
9
10 try {
11     String userName = "root";
12     String password = "root";
13     String url = "jdbc:mysql://localhost:3306/jdbc";
14     Class.forName("com.mysql.jdbc.Driver").newInstance();
15     conn = DriverManager.getConnection(url, userName, password);
16     stmt = conn.createStatement();
17
18     stmt.execute("SELECT * FROM JavaCourse");
19     rs = stmt.getResultSet();
20     System.out.println("\n\n----- Result Before Insert-----\n\n");
21     while (rs.next()) {
22         NameString = rs.getString("Name");
23         RollString = rs.getString("Roll");
24         MarksString = rs.getString("Marks");
25         GradeString = rs.getString("Grade");
26         System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" + Grade: " + GradeString + "\n");
27     } //end while
28
29     stmt.execute("insert into JavaCourse values (01,'Debanish', 75, 'A')");
30     stmt.execute("insert into JavaCourse values (02,'Nilanjan', 85, 'B')");
31     stmt.execute("insert into JavaCourse values (03,'Tauheed', 65, 'B')");
32     stmt.execute("insert into JavaCourse values (04,'Priyabrata', 70, 'A')");
33
34     stmt.execute("SELECT * FROM JavaCourse");
35     rs = stmt.getResultSet();
36     System.out.println("\n\n----- Result After Insert-----\n\n");
37     while (rs.next()) {
38         NameString = rs.getString("Name");
39         RollString = rs.getString("Roll");
40         MarksString = rs.getString("Marks");
41         GradeString = rs.getString("Grade");
42         System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" + Grade: " + GradeString + "\n");
43     } //end while
44 }
```

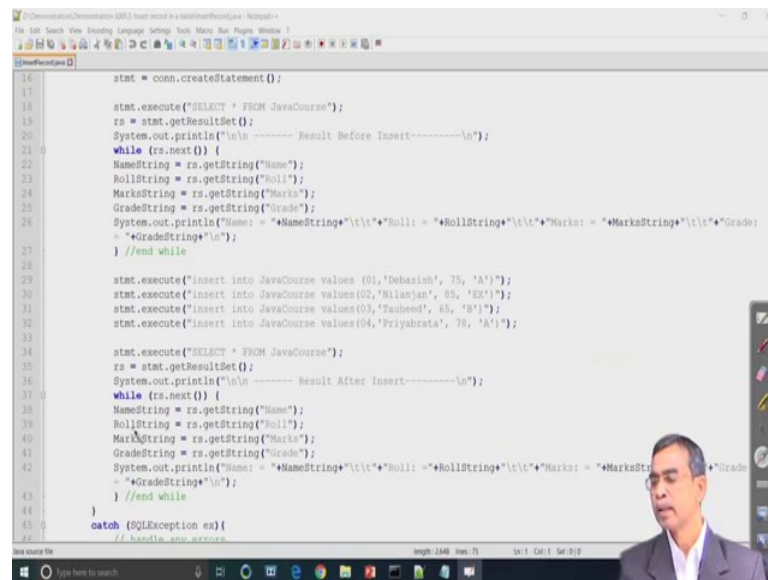
The IDE interface includes a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, Help), a toolbar, and a status bar at the bottom showing 'Line: 1, Col: 1, Sel: 0/0'.

Now here is the course. Here is the obvious course for creating a connection and then loading the JDBC driver and then, here is basically we creating the statement objects and this is the first statement that we are going to create is the statement star from java course. That means, before inserting what is the content it is there, the result sets and this result set will basically display whatever the records that it returns to us and we will see exactly before create before inserting a data, how what are the different records are there.

And then next we fire few statements regarding inserting one record at a time. So, here is inserting first record then second record third record and fourth record. So, if the table is empty then it will show only these four records. If table contents already data so, then all these four newly inserted record will be added after the existing data and then this statement execute select star from JavaCourse.

Basically, same is basically select all the records and get it the result set will receive it and then finally, the result set will here is a while loop the result set will be process.

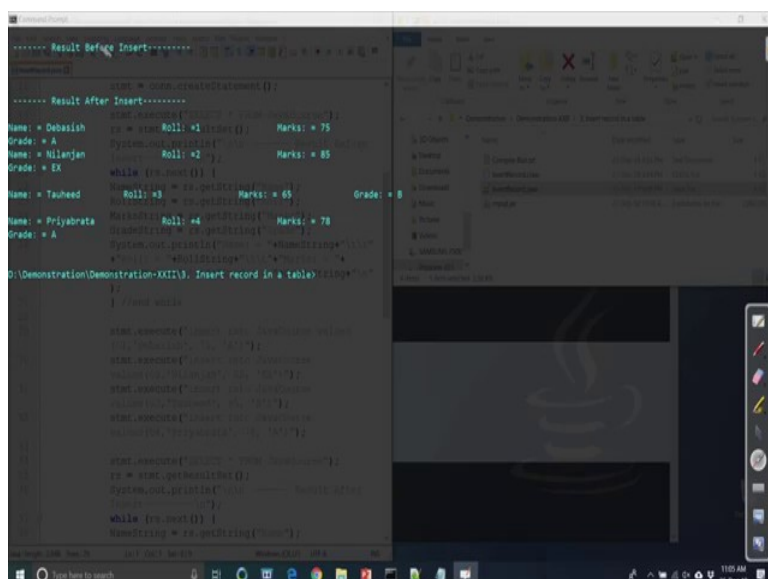
(Refer Slide Time: 14:25)



```
16 stmt = conn.createStatement();
17
18 stmt.execute("SELECT * FROM JavaCourse");
19 rs = stmt.getResultSet();
20 System.out.println("\n\n ----- Result Before Insert-----\n\n");
21 while (rs.next()) {
22     NameString = rs.getString("Name");
23     RollString = rs.getString("Roll");
24     MarksString = rs.getString("Marks");
25     GradeString = rs.getString("Grade");
26     System.out.println("Name: " + NameString + "\t\t\t" + RollString + "\t\t\t" + MarksString + "\t\t\t" + GradeString + "\n");
27 } //end while
28
29 stmt.execute("insert into JavaCourse values (01,'Debanish', 75, 'A')");
30 stmt.execute("insert into JavaCourse values (02,'Nilanjan', 85, 'B')");
31 stmt.execute("insert into JavaCourse values (03,'Tauheed', 65, 'B')");
32 stmt.execute("insert into JavaCourse values (04,'Priyabrata', 78, 'A')");
33
34 stmt.execute("SELECT * FROM JavaCourse");
35 rs = stmt.getResultSet();
36 System.out.println("\n\n ----- Result After Insert-----\n\n");
37 while (rs.next()) {
38     NameString = rs.getString("Name");
39     RollString = rs.getString("Roll");
40     MarksString = rs.getString("Marks");
41     GradeString = rs.getString("Grade");
42     System.out.println("Name: " + NameString + "\t\t\t" + RollString + "\t\t\t" + MarksString + "\t\t\t" + GradeString + "\n");
43 } //end while
44
45 catch (SQLException ex){
46     // handle any errors
47 }
```

And then result set will be processed and it will display all the records that we have receive it. So, this is the program. Now, let us come to the execution of this program, so that you can understand how it is working. So, insert record here.

(Refer Slide Time: 14:41)



```
----- Result Before Insert-----
Name: 
Roll: 
Marks: 
Grade: 

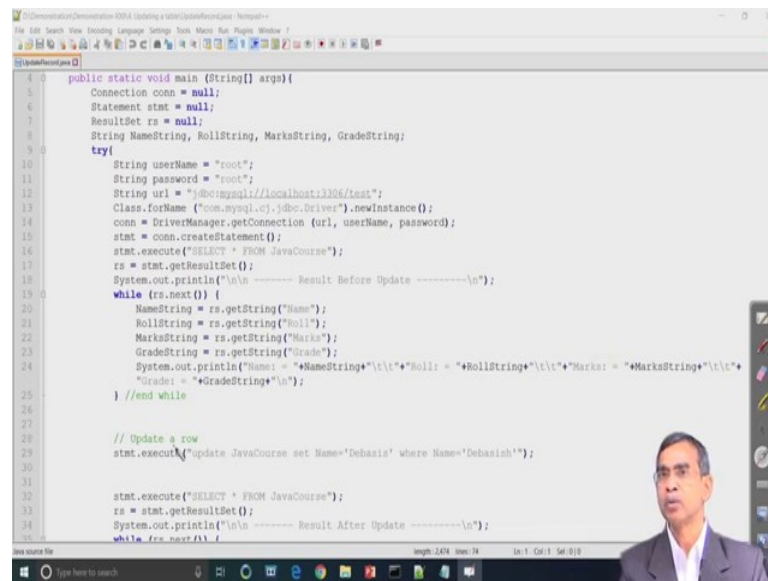
----- Result After Insert-----
Name: Debanish
Roll: 01
Marks: 75
Grade: A
Name: Nilanjan
Roll: 02
Marks: 85
Grade: B
Name: Tauheed
Roll: 03
Marks: 65
Grade: B
Name: Priyabrata
Roll: 04
Marks: 78
Grade: A
```

As you see here Results Before Insert. So, these basically the table does not contain any record. So, it basically empty. Then Result After Insert, as we see this is the record first record we have in inserted this is the second record third record and fourth record. So, the results have been receives and then they have been displayed on the screen; that

means, we can confirm that the results are now available at the application in itself. So, this is the example that is an insert, opposite to the insert our next example is update example.

So, we will just going to update a particular record. As it is always there we will execute the command that what are the records are there and then we will update a record and then after the updation we will see whether updation is reflected or not.

(Refer Slide Time: 15:31)

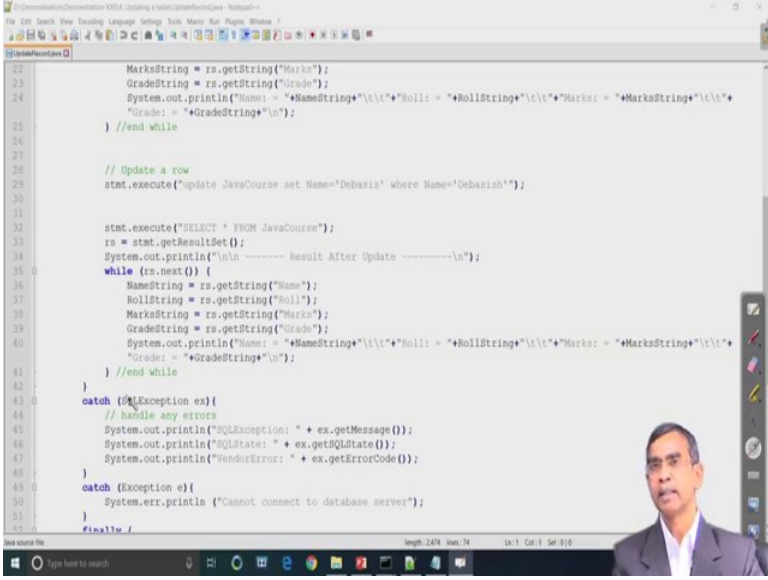


```
4 public static void main (String[] args){
5     Connection conn = null;
6     Statement stmt = null;
7     ResultSet rs = null;
8     String NameString, RollString, MarksString, GradeString;
9     try{
10        String userName = "root";
11        String password = "root";
12        String url = "jdbc:mysql://localhost:3306/test";
13        Class.forName ("com.mysql.cj.jdbc.Driver").newInstance();
14        conn = DriverManager.getConnection (url, userName, password);
15        stmt = conn.createStatement();
16        stmt.execute("SELECT * FROM JavaCourse");
17        rs = stmt.getResultSet();
18        System.out.println("\n\n ----- Result Before Update ----- \n\n");
19        while (rs.next()) {
20            NameString = rs.getString("Name");
21            RollString = rs.getString("Roll");
22            MarksString = rs.getString("Marks");
23            GradeString = rs.getString("Grade");
24            System.out.println("Name: "+NameString+"\t\t\tRoll: "+RollString+"\t\t\tMarks: "+MarksString+"\t\t\t"+
25                               "Grade: "+GradeString+"\n");
26        } //end while
27
28        // Update a row
29        stmt.executeUpdate("update JavaCourse set Name='Debasis' where Name='Debasish'");
30
31        stmt.execute("SELECT * FROM JavaCourse");
32        rs = stmt.getResultSet();
33        System.out.println("\n\n ----- Result After Update ----- \n\n");
34        while (rs.next()) {
35            NameString = rs.getString("Name");
36            RollString = rs.getString("Roll");
37            MarksString = rs.getString("Marks");
38            GradeString = rs.getString("Grade");
39            System.out.println("Name: "+NameString+"\t\t\tRoll: "+RollString+"\t\t\tMarks: "+MarksString+"\t\t\t"+
40                               "Grade: "+GradeString+"\n");
41        } //end while
42    } catch (Exception e) {
43        e.printStackTrace();
44    }
45 }
```

So, we will receive the results after the updation and we will display the updated record on the screen. So, this is the program flow and here is basically creating the connection and then executing the statement that what are the records are there. So, the before updation and then it basically show all the reports and here is the command by which we execute the update. Here we see Update JavaCourse set Name Debasis where Name is Debasish. So, if there is any record whose value is basically Debasis as a Name field and it will basically change into this form. So, this is the next and this is the previous one.

So, it is the updation is there. Only one field value of a record will be updated here. If there are many records containing the same thing, all records in fact, will be updated by this command because it will updated it will scan entire data base tables and then finally, whichever the condition is match it will basically applied to this one, bind to that, records and then do it.

(Refer Slide Time: 16:45)

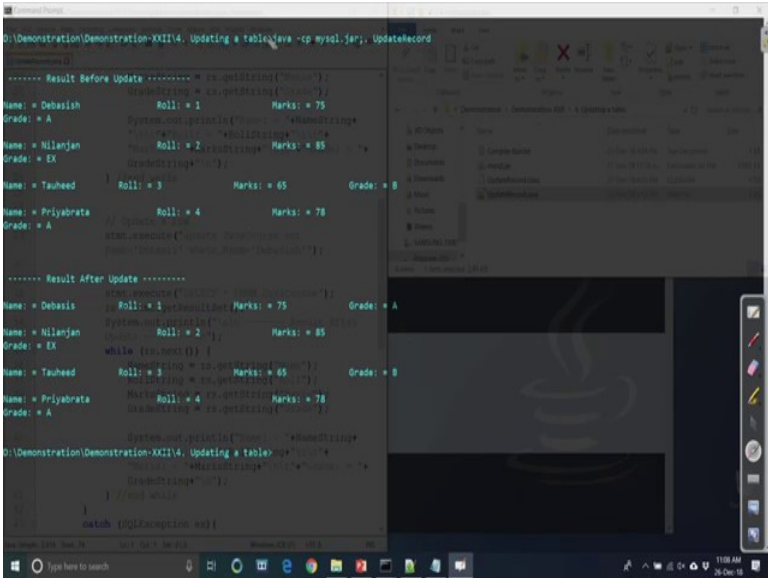


```
22 MarksString = rs.getString("Marks");
23 GradeString = rs.getString("Grade");
24 System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" +
25   "Grade: " + GradeString + "\n");
26 } //end while
27
28 // Update a row
29 stat.execute("update JavaCourse set Name='Debasish' where Name='Debasish'");
30
31 stat.execute("SELECT * FROM JavaCourse");
32 rs = stat.getResultSet();
33 System.out.println("\n\n ----- Result After Update ----- \n\n");
34 while (rs.next()) {
35   NameString = rs.getString("Name");
36   RollString = rs.getString("Roll");
37   MarksString = rs.getString("Marks");
38   GradeString = rs.getString("Grade");
39   System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" +
40     "Grade: " + GradeString + "\n");
41 } //end while
42
43 catch (SQLException ex){
44   // handle any errors
45   System.out.println("SQLException: " + ex.getMessage());
46   System.out.println("SQLState: " + ex.getSQLState());
47   System.out.println("VendorError: " + ex.getErrorCode());
48 }
49 catch (Exception e){
50   System.err.println("Cannot connect to database server");
51 }
52 finally {
53 }
```

And then find and our next step is basically we are going to we are going to again execute that select command as we see here select command Select star from JavaCourse. It basically after the updation, it will receive the results and then all the results are displayed on the screen and then these are the necessary try, catch formalities.

Now, let us have the execution of this program, it is called the update record. This is the name of the program that we have given. We are running this program update record right.

(Refer Slide Time: 17:15)



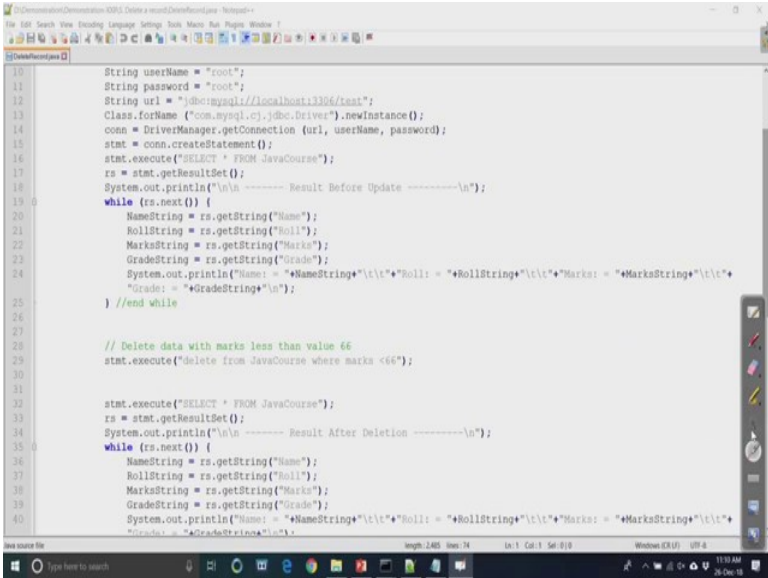
```
..... Result Before Update .....
Name: = Debasish      Roll: = 1      Marks: = 75      Grade: = A
Name: = Nilanjan      Roll: = 2      Marks: = 85      Grade: = EX
Name: = Tauheed       Roll: = 3      Marks: = 65      Grade: = B
Name: = Priyabrata    Roll: = 4      Marks: = 78      Grade: = A

..... Result After Update .....
Name: = Debasish      Roll: = 1      Marks: = 75      Grade: = A
Name: = Nilanjan      Roll: = 2      Marks: = 85      Grade: = EX
Name: = Tauheed       Roll: = 3      Marks: = 65      Grade: = B
Name: = Priyabrata    Roll: = 4      Marks: = 78      Grade: = A
```

And this will show the records before the update. Now here is the execution the total output of the execution. Here you see yeah. So, here as we see start from here, so, this is the output started from the screen here after executing the update record. So, Results Before Update as we see Debasis, Nilanjan, Tauheed, Priyabrata; four records are there.

Now, we have executed update command. Then results after update command as you see initially, Debasis has been updated Debasis and it is only applicable to this record. So, it is only one updation is there and then the result is there. So, now, we can understand that. So, everything is possible from your program itself right and then all the SQL statement either execution then after getting result after the result is available processing all those results from the one point of node only.

(Refer Slide Time: 18:17)

The image is a screenshot of a Java IDE window titled "Delete a record DemoRecord.java - NetBeans". The code in the editor defines a Java program that connects to a MySQL database, retrieves records from a table named 'JavaCourse', prints them, deletes records with marks less than 66, and then prints the remaining records. The output window at the bottom shows the execution results, including the records before deletion and the records after deletion. The records before deletion are: Debasis, Nilanjan, Tauheed, and Priyabrata. The records after deletion are: Debasis, Nilanjan, and Tauheed. The record for Priyabrata has been successfully deleted.

```
10 String userName = "root";
11 String password = "root";
12 String url = "jdbc:mysql://localhost:3306/test";
13 Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
14 conn = DriverManager.getConnection(url, userName, password);
15 stmt = conn.createStatement();
16 stmt.execute("SELECT * FROM JavaCourse");
17 rs = stmt.getResultSet();
18 System.out.println("\n\n----- Result Before Update -----");
19 while (rs.next()) {
20     NameString = rs.getString("Name");
21     RollString = rs.getString("Roll");
22     MarksString = rs.getString("Marks");
23     GradeString = rs.getString("Grade");
24     System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" +
25         "Grade: " + GradeString + "\n");
26 } //end while
27
28 // Delete data with marks less than value 66
29 stmt.execute("Delete from JavaCourse where marks < 66");
30
31
32 stmt.execute("SELECT * FROM JavaCourse");
33 rs = stmt.getResultSet();
34 System.out.println("\n\n----- Result After Deletion -----");
35 while (rs.next()) {
36     NameString = rs.getString("Name");
37     RollString = rs.getString("Roll");
38     MarksString = rs.getString("Marks");
39     GradeString = rs.getString("Grade");
40     System.out.println("Name: " + NameString + "\t\t" + Roll: " + RollString + "\t\t" + Marks: " + MarksString + "\t\t" +
41         "Grade: " + GradeString + "\n");
42 }
```

Java course file length: 2.4Kb lines: 74 in: 1 Col: 1 Sel: 0 Windows (X) 0 UTF-8

Output:

```
----- Result Before Update -----
Name: Debasis \t\t Roll: 1 \t\t Marks: 88 \t\t Grade: A
Name: Nilanjan \t\t Roll: 2 \t\t Marks: 75 \t\t Grade: B
Name: Tauheed \t\t Roll: 3 \t\t Marks: 92 \t\t Grade: A+
Name: Priyabrata \t\t Roll: 4 \t\t Marks: 65 \t\t Grade: C
----- Result After Deletion -----
Name: Debasis \t\t Roll: 1 \t\t Marks: 88 \t\t Grade: A
Name: Nilanjan \t\t Roll: 2 \t\t Marks: 75 \t\t Grade: B
Name: Tauheed \t\t Roll: 3 \t\t Marks: 92 \t\t Grade: A+
-----
```

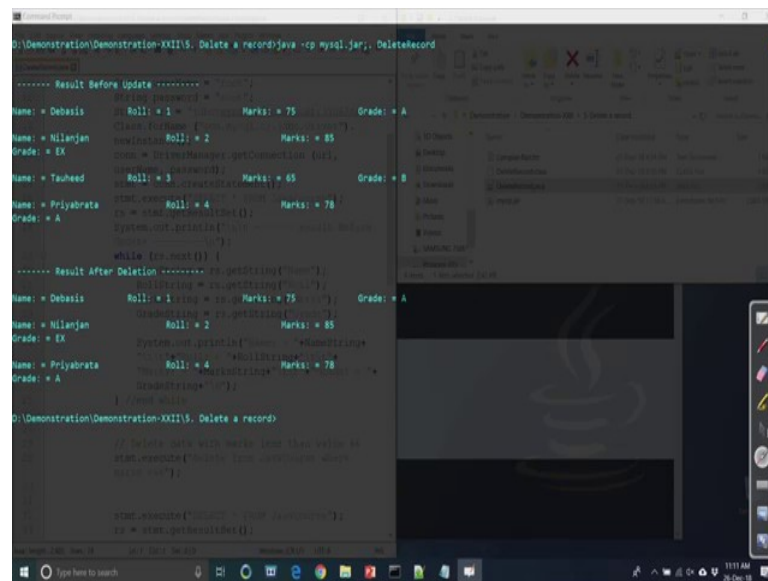
Now, our next example; this is the example ok, so, for the delete is common. So, that whether delete command can be also successfully executed or not. So, first in the same manner we will first see all the tables; accountants, first records and then we will issue a delete command and after the delete command we will exactly see the results.

So, here is the program flow, creating the connection. Then we execute the first statement; Select star from JavaCourse is basically receive all the records and then display all the results on the screen after the execution of the statement. Then we execute the delete command; delete from java course where marks less than 66. If there are any record records or there is any there is a record whose value of the field marks is less than

66 will be deleted from the record and after the execution of this statement we will be able to again see the records after the deletion.

So, we have to issue another command Select star from JavaCourse, it will get all the results and then there result will be printed on the screen. The result will be printed on the screen and then rest of the program is basically try catch. Now, let us run this program, this program is a delete record.

(Refer Slide Time: 19:41)

The screenshot shows a Windows command prompt window with a Java program running. The program displays the state of a database before and after deleting a record. Before deletion, there are four records: Debasis (Roll: 1, Marks: 75, Grade: A), Nilanjan (Roll: 2, Marks: 85, Grade: B), Tauheed (Roll: 3, Marks: 65, Grade: B), and Priyabrata (Roll: 4, Marks: 78, Grade: A). After the deletion command is executed, the output shows only three records: Debasis, Nilanjan, and Priyabrata. The record for Tauheed (Roll: 3) has been removed. The program also includes a try-catch block for handling exceptions.

```

D:\Demonstration\Demonstration-KXII\15. Delete a record>java -cp mysql.jar;. DeleteRecord

----- Result Before Update -----
Name = Debasis      Roll: = 1      Marks: = 75      Grade: = A
Name = Nilanjan     Roll: = 2      Marks: = 85
Name = Tauheed      Roll: = 3      Marks: = 65      Grade: = B
Name = Priyabrata   Roll: = 4      Marks: = 78      Grade: = A

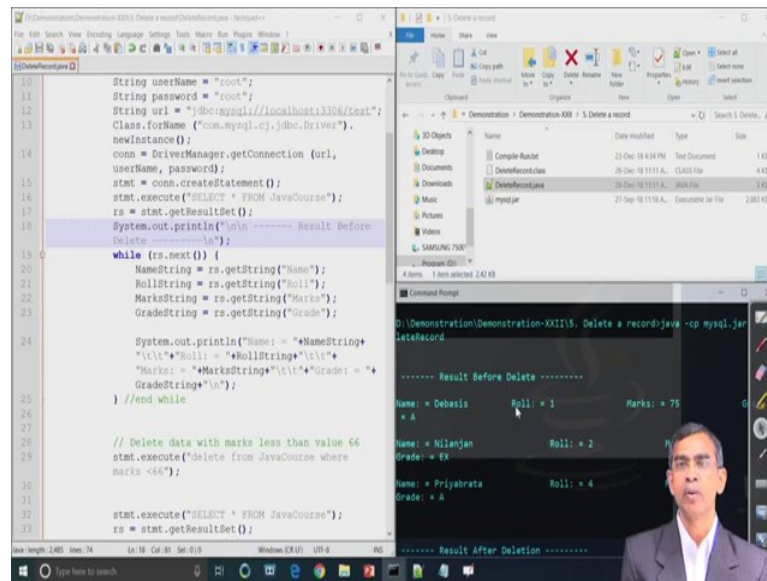
----- Result After Deletion -----
Name = Debasis      Roll: = 1      Marks: = 75      Grade: = A
Name = Nilanjan     Roll: = 2      Marks: = 85
Name = Priyabrata   Roll: = 4      Marks: = 78      Grade: = A

D:\Demonstration\Demonstration-KXII\15. Delete a record>
```

And this is the execution results written results. As we see delete record is executed here. So, Results Before Delete ok; result before delete any way. So, there is a mistakes there in the program itself is updated. Anyway, so, you read add delete and then these are the four records as we see. Results after deletion as we see here actually the marks is 75, 85, 65, 78.

So, if the execution is correct then the third records should be deleted from the table. As you see the first record is there, second record is there, fourth record is there and third record is no more. So, it has been deleted. So, we have learned about how the result set object is there.

(Refer Slide Time: 20:25)



Now, so far the JDBC programming is concerned there are few steps that we have learnt about and all the steps that we have done it and then also the sufficient demonstration we have given starting from how the MySQL server can be installed.

And then how the MySQL server can be used using the console based MySQL server accessing. Then connection to the server using the JDBC and how the JDBC driver also can be downloaded and it can be archived and then the command can be executed. And then we can after the execution of command how the results can be obtained or receives into the program and the program can process these things. So, this is the demo about the our last third say part of the demo, so for the JDBC is concerned and this is the end of our demonstration session as well as all the demonstration.

In this course we have covered around 20 22 demonstration and then other around 28 lecture slice in other. So, basically almost 50 percent demonstration followed by the 50 percent theoretical discussion, so that all the theoretically leaned concept can be complimented through the demonstration.

I hope you have enjoyed the course and the demonstration. In our next section, we will discuss about some java projects that projects can be handled by you as a single team member or as a jointly with many team people involved. Now we will discuss the projects and then we will basically see exactly how far you have learnt and then whether we can solve any real life application sort of things in this ok.

Thank you very much.