Pune Institute of Computer Technology, Pune

Department of Computer Engineering

A.Y. 2020-21 Semester: I

Database Management System Lab

Name: Aditya Sawant

Roll No: 31302

Batch: K3

ASSIGNMENT 2

TITLE:

Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as creation of (Table, View, Index, Sequence, Synonym).

PROBLEM DEFINITION:

Implement DDL commands in context of view, index, sequence using JDBC.

OBJECTIVE:

- To understand implementation of JDBC.
- To understand & implement the various DDL and DML Commands.
- To understand database concepts like view, index ,sequence and synonym.

OUTCOME:

We will be able to understand and implement DDL and DML commands using JDBC.

HARDWARE REQUIREMENTS:

- MONITOR
- KEYBOARD
- 2GB RAM
- 2.4GHz I5 PROCESSOR

SOFTWARE REQUIREMENTS:

- DATABASE-MYSQL
- IDE-ECLIPSE
- OS-FEDORA 20

Theory:

JDBC:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

STEPS INVOLVED IN JDBC:

The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:

1. Import JDBC packages.
2. Load and register the JDBC driver.
3. Open a connection to the database.
4. Create a statement object to perform a query.
5. Execute the statement object and return a query resultset.
6. Process the resultset.
7. Close the resultset and statement objects.
8. Close the connection.

Applications of JDBC:

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as −

- Java Applications
- Java Applets
- Java Servlets
- Java ServerPages (JSPs)
- Enterprise JavaBeans (EJBs).

INDEX:

An index can be created in a table to find data more quickly and efficiently. The users cannot see the indexes, they are just used to speed up searches/queries.

SYNTAX-

CREATE INDEX index_name ON table_name (column_name);

SEQUENCE:

A sequence is a set of integers 1, 2, 3, ... that are generated in order on a specific demand. Sequences are frequently used in the databases because many applications require each row in a table to contain a unique value and sequences provide an easy way to generate them.

The simplest way in MySQL to use Sequences is to define a column as AUTO_INCREMENT and leave the remaining things to MySQL to take care.

SYNTAX-

CREATE Sequence sequence-name start with initial-value increment by increment-value maxvalue maximum-value cycle|nocycle

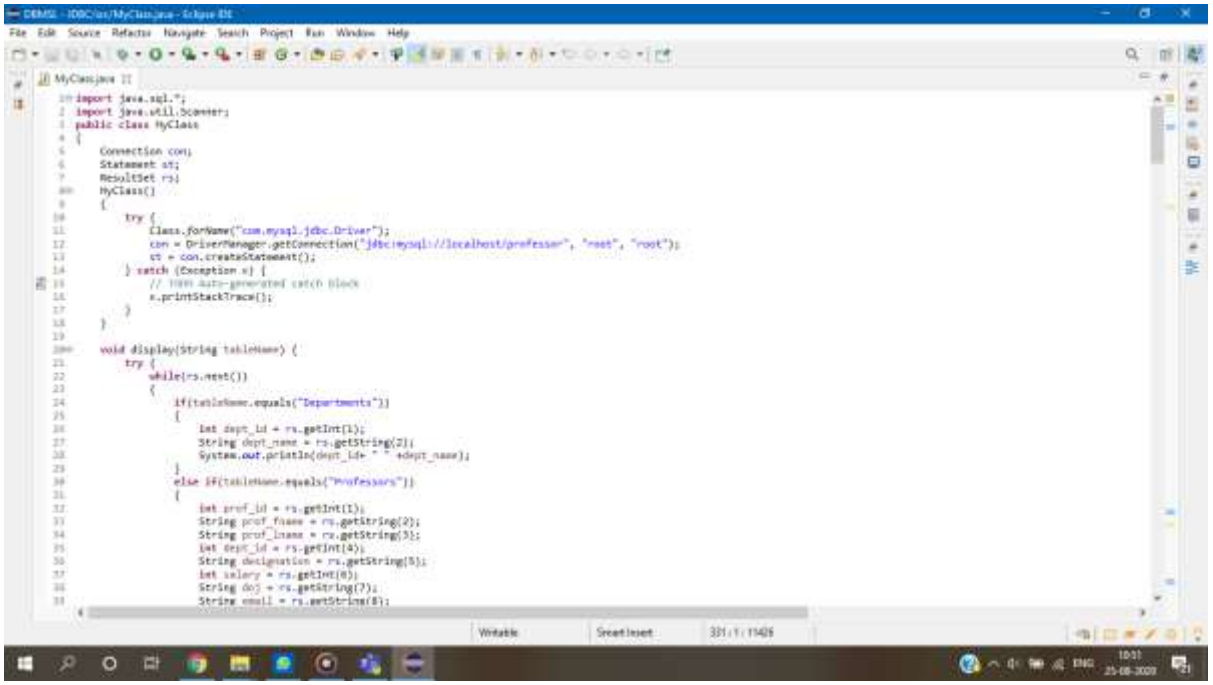
SYNONYM:

Synonyms provide both data independence and location transparency. Synonyms permit applications to function without modification regardless of which user owns the table or view and regardless of which database holds the table or view. However, synonyms are not a substitute for privileges on database objects. Appropriate privileges must be granted to a user before the user can use the synonym.

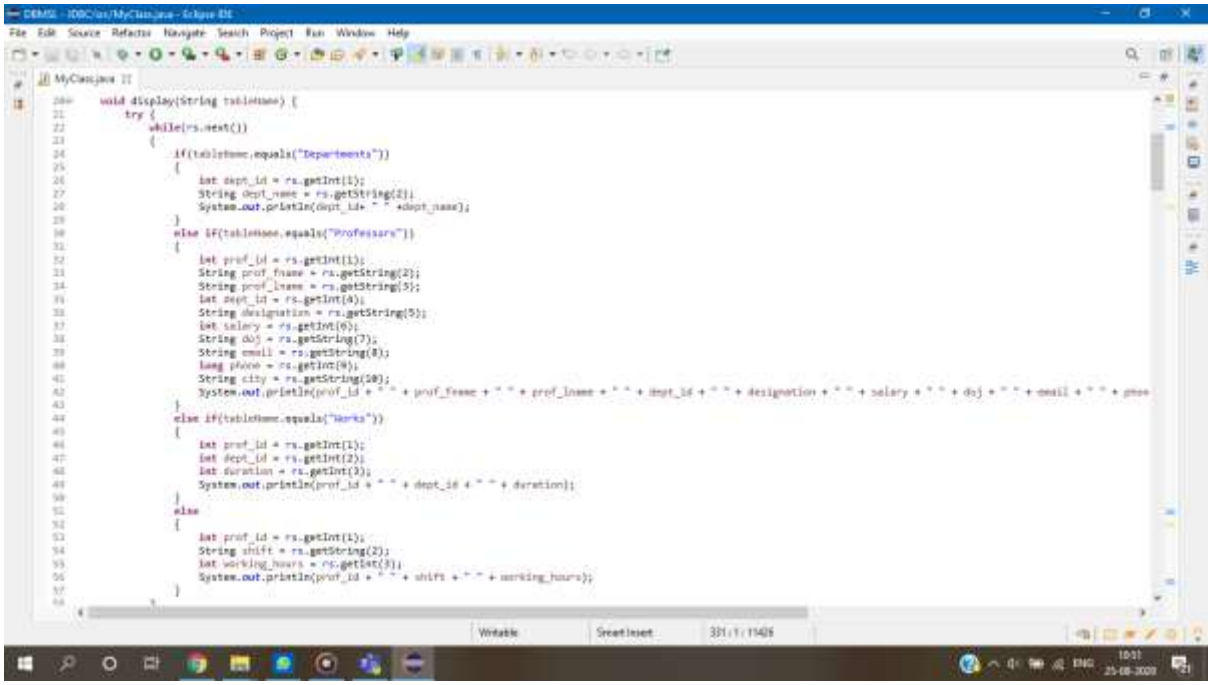
SYNTAX-

Use the CREATE SYNONYM statement to create a synonym

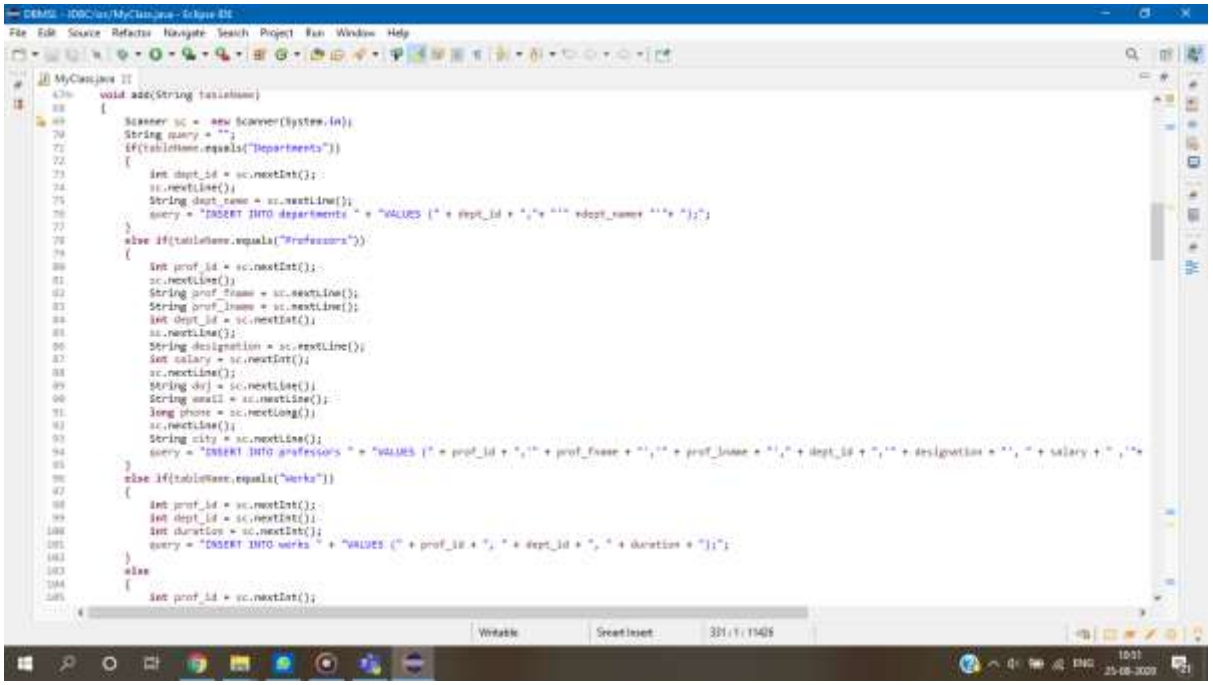
Code:



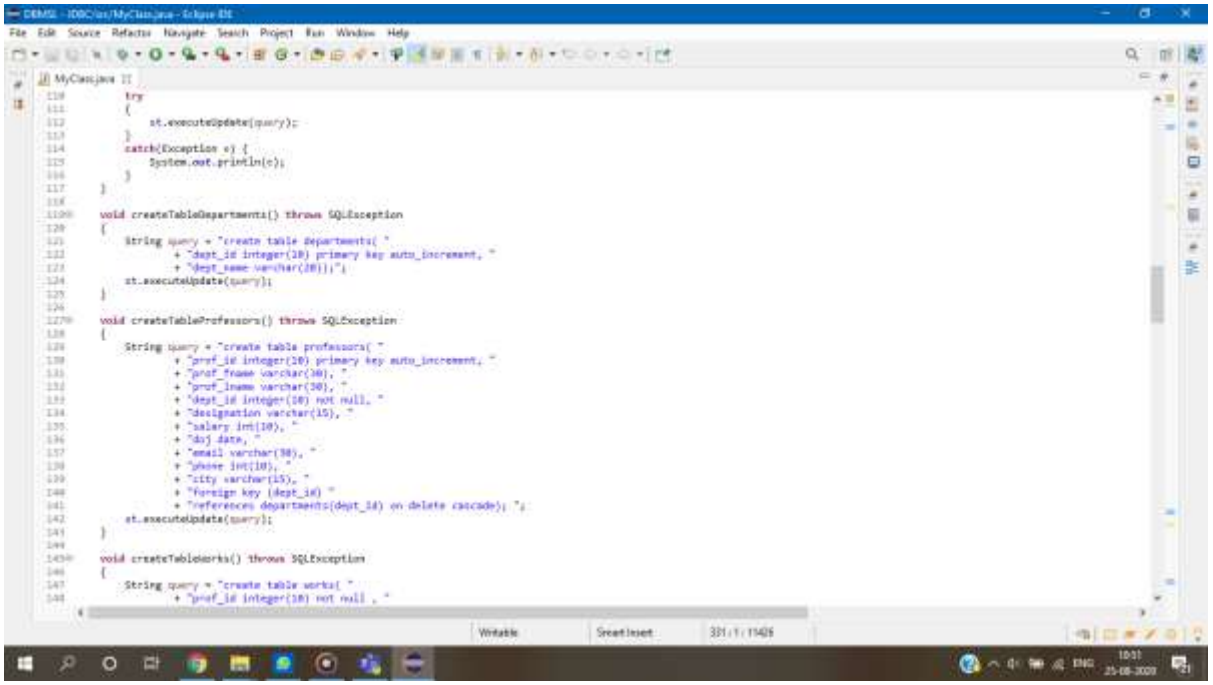
```
import java.sql.*;
import java.util.Scanner;
public class MyClass
{
    Connection con;
    Statement st;
    ResultSet rs;
    MyClass()
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql:///localhost/professor", "root", "root");
            st = con.createStatement();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    void display(String tableName) {
        try {
            while(rs.next())
            {
                if(tableName.equals("Departments"))
                {
                    int dept_id = rs.getInt(1);
                    String dept_name = rs.getString(2);
                    System.out.println(dept_id+ " " +dept_name);
                }
                else if(tableName.equals("Professors"))
                {
                    int prof_id = rs.getInt(1);
                    String prof_fname = rs.getString(2);
                    String prof_lname = rs.getString(3);
                    int dept_id = rs.getInt(4);
                    String designation = rs.getString(5);
                    int salary = rs.getInt(6);
                    String doj = rs.getString(7);
                    String email = rs.getString(8);
```



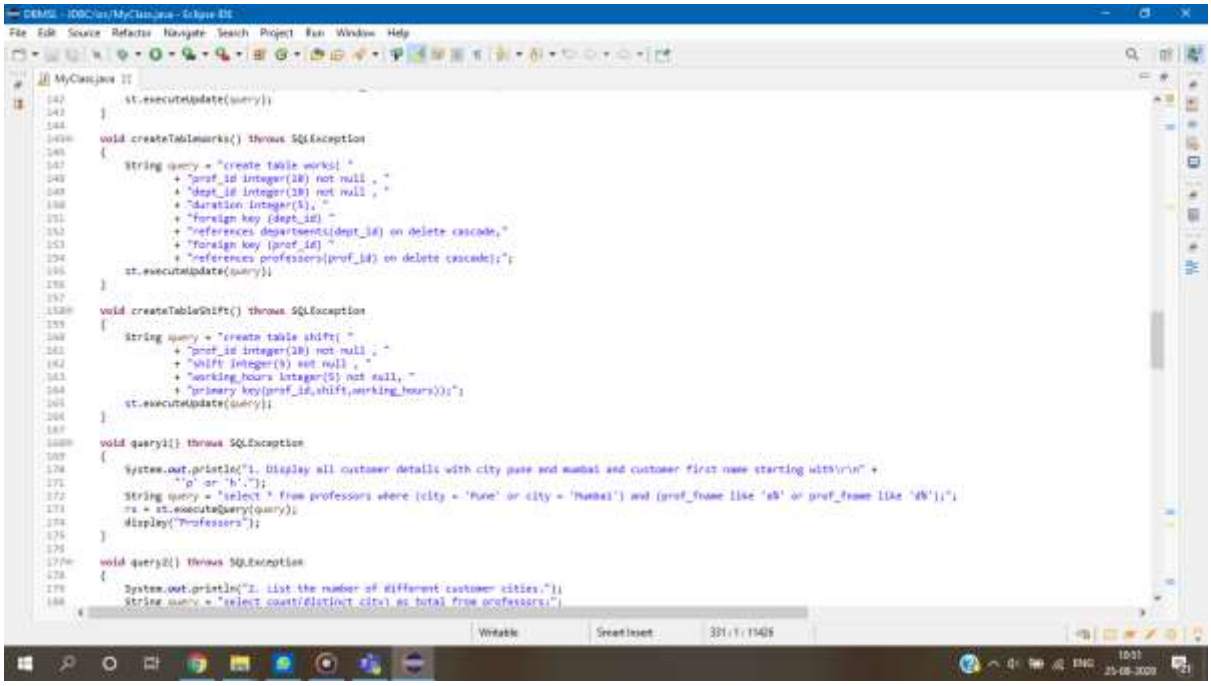
```
    void display(String tableName) {
        try {
            while(rs.next())
            {
                if(tableName.equals("Departments"))
                {
                    int dept_id = rs.getInt(1);
                    String dept_name = rs.getString(2);
                    System.out.println(dept_id+ " " +dept_name);
                }
                else if(tableName.equals("Professors"))
                {
                    int prof_id = rs.getInt(1);
                    String prof_fname = rs.getString(2);
                    String prof_lname = rs.getString(3);
                    int dept_id = rs.getInt(4);
                    String designation = rs.getString(5);
                    int salary = rs.getInt(6);
                    String doj = rs.getString(7);
                    String email = rs.getString(8);
                    long phone = rs.getInt(9);
                    String city = rs.getString(10);
                    System.out.println(prof_id + " " + prof_fname + " " + prof_lname + " " + dept_id + " " + designation + " " + salary + " " + doj + " " + email + " " + phon
                }
                else if(tableName.equals("Works"))
                {
                    int prof_id = rs.getInt(1);
                    int dept_id = rs.getInt(2);
                    int duration = rs.getInt(3);
                    System.out.println(prof_id + " " + dept_id + " " + duration);
                }
                else
                {
                    int prof_id = rs.getInt(1);
                    String shift = rs.getString(2);
                    int working_hours = rs.getInt(3);
                    System.out.println(prof_id + " " + shift + " " + working_hours);
                }
```



```
void add(String tableName)
{
    Scanner sc =  new Scanner(System.in);
    String query = "";
    if(tableName.equals("Departments"))
    {
        int dept_id = sc.nextInt();
        sc.nextLine();
        String dept_name = sc.nextLine();
        query = "INSERT INTO departments " + "VALUES (" + dept_id + ","+ "'" +dept_name+ "'"+ ");";
    }
    else if(tableName.equals("Professors"))
    {
        int prof_id = sc.nextInt();
        sc.nextLine();
        String prof_fname = sc.nextLine();
        String prof_lname = sc.nextLine();
        int dept_id = sc.nextInt();
        sc.nextLine();
        String designation = sc.nextLine();
        int salary = sc.nextInt();
        sc.nextLine();
        String doj = sc.nextLine();
        String email = sc.nextLine();
        long phone = sc.nextLong();
        sc.nextLine();
        String city = sc.nextLine();
        query = "INSERT INTO professors " + "VALUES (" + prof_id + ",'" + prof_fname + "','" + prof_lname + "'," + dept_id + ",'" + designation + "', " + salary + " ,'"+
    }
    else if(tableName.equals("Works"))
    {
        int prof_id = sc.nextInt();
        int dept_id = sc.nextInt();
        int duration = sc.nextInt();
        query = "INSERT INTO works " + "VALUES (" + prof_id + ", " + dept_id + ", " + duration + ");";
    }
    else
    {
        int prof_id = sc.nextInt();
```

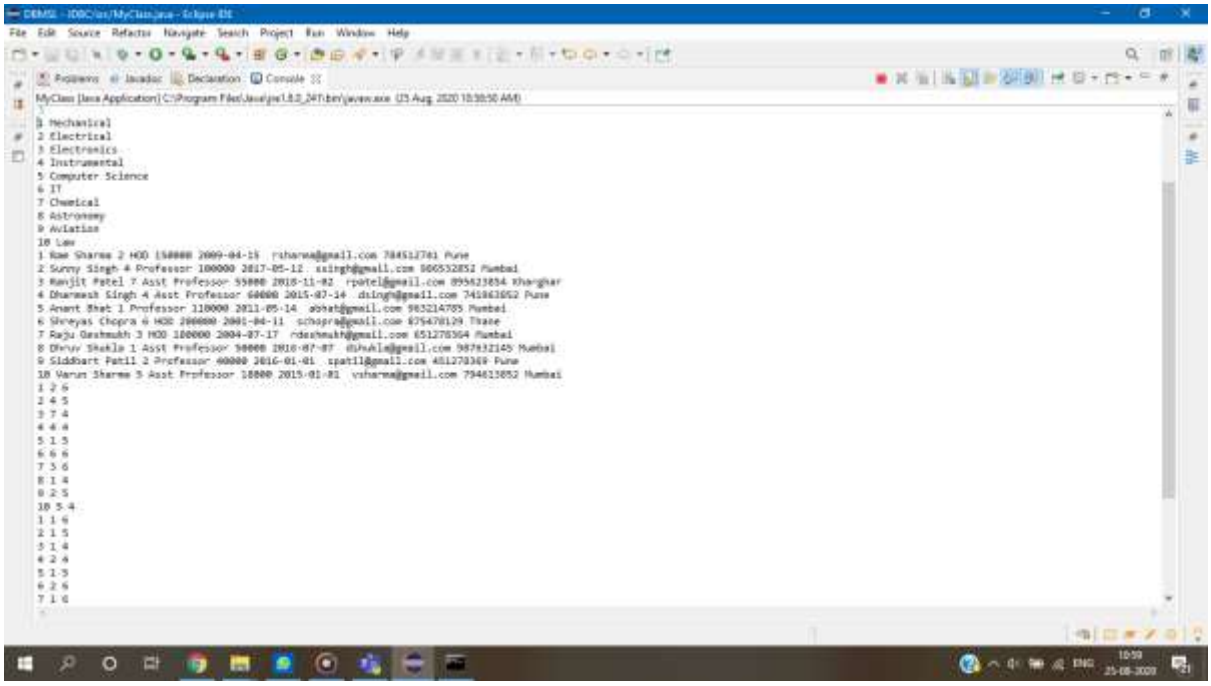


```
        try
        {
            st.executeUpdate(query);
        }
        catch(Exception e) {
            System.out.println(e);
        }
    }

    void createTableDepartments() throws SQLException
    {
        String query = "create table departments( "
                + "dept_id integer(10) primary key auto_increment, "
                + "dept_name varchar(20));";
        st.executeUpdate(query);
    }

    void createTableProfessors() throws SQLException
    {
        String query = "create table professors( "
                + "prof_id integer(10) primary key auto_increment, "
                + "prof_fname varchar(30), "
                + "prof_lname varchar(30), "
                + "dept_id integer(10) not null, "
                + "designation varchar(15), "
                + "salary int(10), "
                + "doj date, "
                + "email varchar(30), "
                + "phone int(10), "
                + "city varchar(15), "
                + "foreign key (dept_id) "
                + "references departments(dept_id) on delete cascade); ";
        st.executeUpdate(query);
    }

    void createTableWorks() throws SQLException
    {
        String query = "create table works( "
                + "prof_id integer(10) not null , "
```

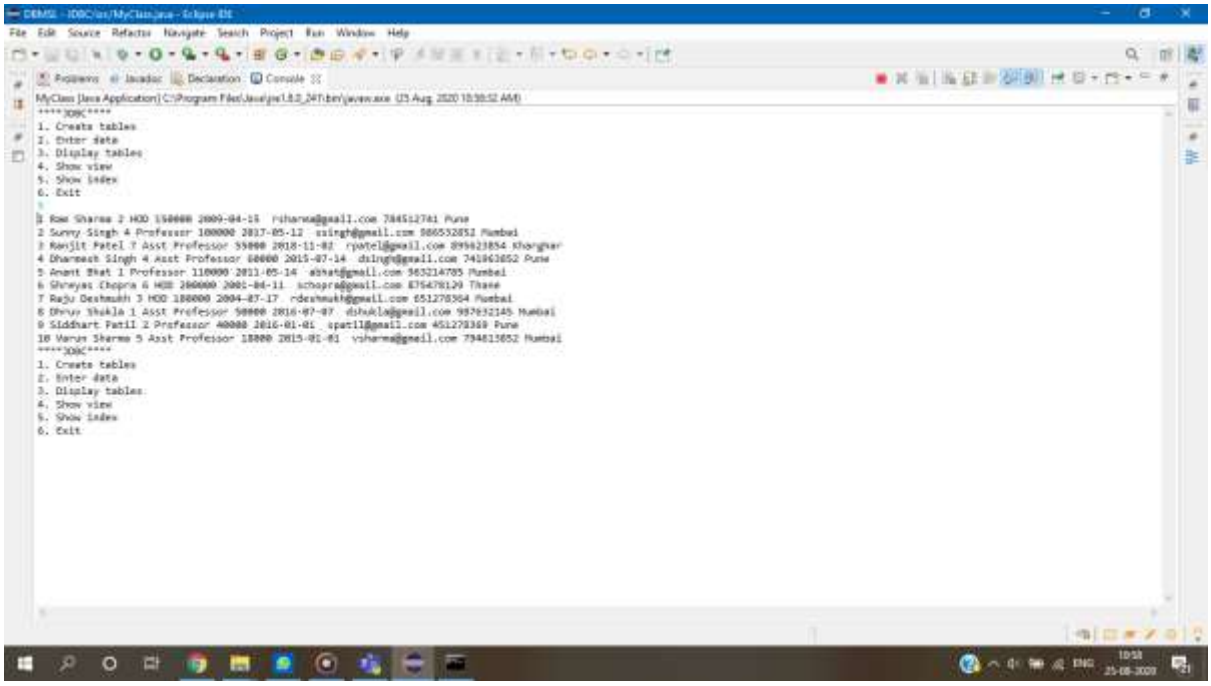


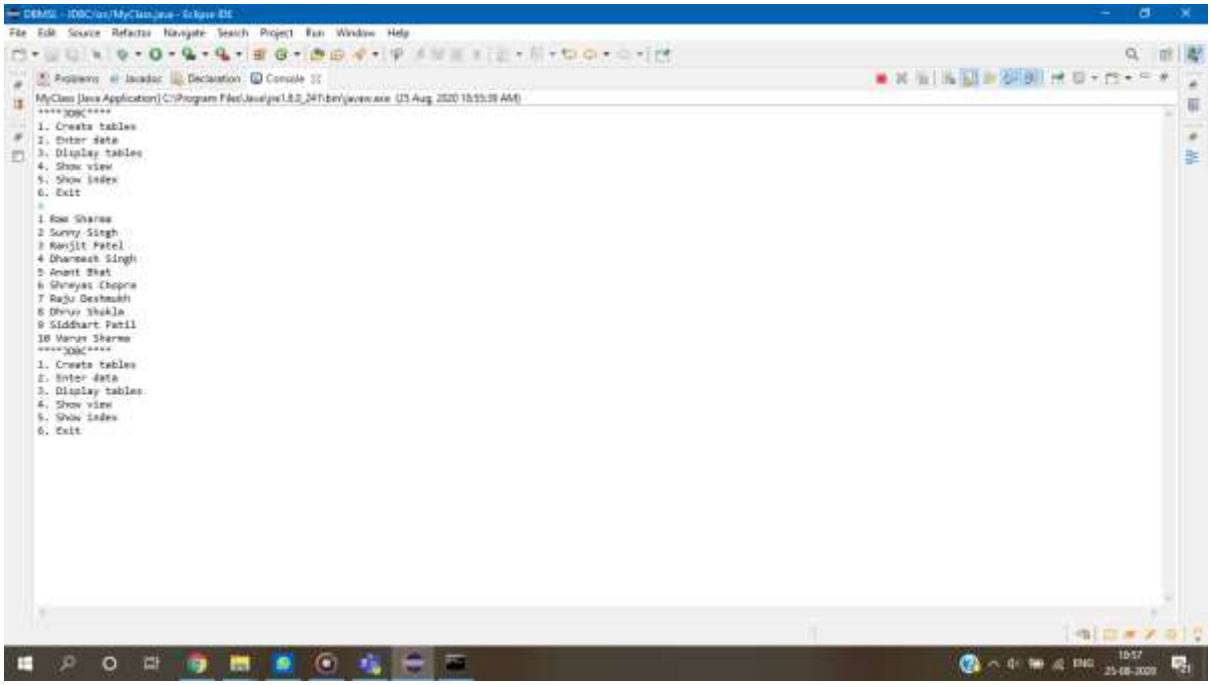
```
        st.executeUpdate(query);
    }

    void createTableworks() throws SQLException
    {
        String query = "create table works( "
                + "prof_id integer(10) not null , "
                + "dept_id integer(10) not null , "
                + "duration integer(5), "
                + "foreign key (dept_id) "
                + "references departments(dept_id) on delete cascade,"
                + "foreign key (prof_id) "
                + "references professors(prof_id) on delete cascade);";
        st.executeUpdate(query);
    }

    void createTableShift() throws SQLException
    {
        String query = "create table shift( "
                + "prof_id integer(10) not null , "
                + "shift integer(5) not null , "
                + "working_hours integer(5) not null, "
                + "primary key(prof_id,shift,working_hours));";
        st.executeUpdate(query);
    }

    void query1() throws SQLException
    {
        System.out.println("1. Display all customer details with city pune and mumbai and customer first name starting with\r\n" +
                "'p' or 'b'.");
        String query = "select * from professors where (city = 'Pune' or city = 'Mumbai') and (prof_fname like 'a%' or prof_fname like 'd%');";
        rs = st.executeQuery(query);
        display("Professors");
    }

    void query2() throws SQLException
    {
        System.out.println("2. List the number of different customer cities.");
        String query = "select count(distinct city) as total from professors;";
```

Outputs:



```
3 Mechanical
2 Electrical
3 Electronics
4 Instrumental
5 Computer Science
6 IT
7 Chemical
8 Astronomy
9 Aviation
10 Law
1 Rae Sharma 2 HOD 150000 2009-04-15  rsharma@gmail.com 784512781 Pune
2 Sunny Singh 4 Professor 100000 2017-05-12  ssingh@gmail.com 986552852 Mumbai
3 Ranjit Patel 7 Asst Professor 55000 2018-11-02  rpatel@gmail.com 895623854 Kharghar
4 Dharmesh Singh 4 Asst Professor 60000 2015-07-14  dsingh@gmail.com 741563852 Pune
5 Anant Bhat 1 Professor 110000 2011-05-14  abhat@gmail.com 963214785 Mumbai
6 Shreyas Chopra 6 HOD 200000 2001-04-11  schopra@gmail.com 8754478129 Thane
7 Raju Deshmukh 3 HOD 100000 2004-07-17  rdeshmukh@gmail.com 651278364 Mumbai
8 Dhruv Shukla 1 Asst Professor 50000 2016-07-07  dshukla@gmail.com 987632145 Mumbai
9 Siddhart Patil 2 Professor 40000 2016-01-01  spatil@gmail.com 451278369 Pune
10 Varun Sharma 5 Asst Professor 18000 2015-01-01  vsharma@gmail.com 794613852 Mumbai
1 2 6
2 4 5
3 7 4
4 4 4
5 1 5
6 6 6
7 5 6
8 1 4
9 2 5
10 5 4
1 1 6
2 1 5
3 1 4
4 2 4
5 1 5
6 2 6
7 1 6
```





CONCLUSION:

WE HAVE SUCCESSFULLY UNDERSTOOD CONCEPTS OF VIEW, INDEXING , SEQUENCE AND SYNONYM AND IMPLEMENTED DDL AND DML COMMANDS USING JDBC.