Java Connection To MySQL

- Before Connection first add JDBC connector Jar file to libraries.
- Make a project in java (NetBeans).
- Add a class and name it dbconnect.

```
import java.sql.*;
public class dbconnect {
    private Connection con; //connection type variable
    private Statement st;
    public PreparedStatement stt; //for prepared statement used later
    private ResultSet rs;
    public dbconnect()
    {try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","");
             st = con.createStatement();
            System.out.println("connected");
             }catch(ClassNotFoundException | SQLException ex){
        System.out.println("error :"+ ex);
    } }
```

```
//In Java Main Class  make an object of dbconnect class to call its
constructor

import java.sql.SQLException;

public class main {


   public static void main(String[] args) throws SQLException {
     dbconnect connect =new dbconnect();
}
```

Creating Database

```
public void create_database () throws SQLException
   {
       String query;
       query="CREATE DATABASE IF NOT EXISTS customers_db";
       st.executeUpdate(query);

       System.out.println("database has successfully been created");
   }
//Java main Class call this method to see output
import java.sql.SQLException;
public class main {


   public static void main(String[] args) throws SQLException {
     dbconnect connect =new dbconnect();
     connect.create_database();
}
```

Creating Tables

```
public void create_table () throws SQLException
   {

       String query = "CREATE TABLE IF NOT EXISTS
customers_db.customers("
             + "customer_id INT(5) NOT NULL AUTO_INCREMENT,"
             + "name VARCHAR(45) NOT NULL,"
             + "email VARCHAR(50) DEFAULT NULL,"
             + "PRIMARY KEY (customer_id)"
             + ");";

       st.executeUpdate(query);

       System.out.println("Customers table has successfully been created");
   }
// In Java main class call this method
```

Inserting Data

```
public void insert_data() throws SQLException
    {

        st.executeUpdate("INSERT INTO persons " + "VALUES (3, 'Hira',23)");
        System.out.println("inserted");

    }
```

Inserting Data Through Prepared Statements

```
public void insert_Prepared_Statement() throws SQLException
 {
    String query;
    query="insert into persons (id,name,age )" +"VALUES(?, ?, ?)";
   stt=con.prepareStatement(query);
  //stt is the prepared statement type variable declare in the class
   stt.setInt(1,9);
   stt.setString(2, "Ayesha");
   stt.setInt(3,25);
   stt.setInt(1,7);
   stt.setString(2,"Aniqa");
   stt.setInt(3,56);
   stt.execute();
   stt.close();
      }
```

Getting Data From Database

```
public void getdata() throws SQLException
    {

        String query;
        query = "SELECT * FROM PERSONS";
        rs=st.executeQuery(query);
        System.out.println("Records from database:");
        while(rs.next()){
          String name=rs.getString("name");
         //’name’ is the attribute of table persons
          String age=rs.getString("age");
         //rs is the result set variable declare in class
          System.out.println(name);
          System.out.println(age);

        }
    }
```

Updating and Deleting Data

```
 public void Update_data() throws SQLException
  {
    System.out.print("\n[Performing UPDATE] ... ");
    st.executeUpdate("UPDATE persons SET age=15 WHERE id=2");

  }

  public void Delete_data() throws SQLException
  {
    System.out.print("\n[Performing DELETE] ... ");
    st.executeUpdate("DELETE FROM persons WHERE NAME=‘Aniqa'");


}
```