

Java Database connectivity :-

Java database connectivity is an application programming interface (API) for the Java programming language which define how a client can access a database. It is a Java-based data access technology used for Java database connectivity.

Types of JDBC drivers :-

Type 1: Type 1 drivers are "bridge" drivers.

Type 2: Type 2 drivers use a native API to communicate with a database system.

Type 3: These drivers use a network protocol and middleware to communicate with a server.

Type 4: Type 4 driver uses Java to implement a DBMS vendor networking protocol.

```
import java . sql . connection;
```

```
import java . sql . DriverManager;
```

```
import java . sql . SQLException;
```

```
import java . sql . Statement;
```

```
import java . util . Scanner;
```

```
Public class main {
```

```
    Public static void main (String [] args){
```

```
        Scanner Scanner = new Scanner(System.in);
```

```
        try {
```

```
            //  
            Class.forName("com . mysql . cj . Jdbc .  
                driver");
```

```
            String url = "Jdbc . mysql ;
```

```
                String user = "root";
```

```
                String Password = " ";
```

```
                Connection con = DriverManager .
```

```
                    get Connection (url ,  
                        user);
```

```
                Statement stmt = con . create Statement ();
```

```
                String create Table SQL = "(CREATE  
                    TABLE IF NOT EXISTS student(Sno  
                        int SNAME - VARCHAR (20));
```



```
For (int i = 0; i < 5; i++) {  
    System.out.println("Enter Student  
number:");  
    int sno = scanner.nextInt();  
    scanner.nextLine();
```

```
    System.out.print("Enter student name:");  
    String sname = scanner.nextLine();
```

```
    String insertSql = String.format("INSERT  
INTO student(sno, sname) values(?,?)");
```

```
}
```

```
System.out.println("Records inserted  
successfully...");
```

```
stmt.close();
```

```
con.close();
```

```
} catch (ClassNotFoundException | SQLException  
e) {  
    e.printStackTrace();
```

```
} finally {  
    scanner.close();
```

```
}
```

```
}
```

```
}
```

```
import java.sql.Connection  
import java.sql.DriverManager  
import java.sql.Exception;  
import java.sql.Statement;  
import java.sql;  
import java.util.Scanner;
```

```
Public class main {  
    Public static void main (String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        try
```

```
        class.forName("com.mysql.jdbc.Driver");
```

```
        String url = "jdbc:mysql";
```

```
        String user = "root";
```

```
        String Password = "";
```

```
        Connection con = DriverManager.getConnection(url, user, Password);
```

```
        Statement stmt = con.createStatement();
```

```
        String createTableSQL = "CREATE
```

```
TABLE IF NOT EXISTS
```

```
student (SNO int, SNAME VARCHAR(20),
```

```

for (int i = 0; i < 5; i++) {
    System.out.print("Enter student number ");
    int sno = scanner.nextInt();
    scanner.nextLine();

    System.out.println("Enter student name ");
    String sname = scanner.nextLine();

    String insertSql = String.format("INSERT INTO student(SNO, SNAME) VALUES (%d, '%s')",
    sno, sname);

    stmt.executeUpdate(insertSql);
}

System.out.println("Records inserted successfully ...");

String sql = "SELECT * FROM student";
ResultSet rs = null;
rs = stmt.executeQuery(sql);
System.out.println("Student details are:");
while (rs.next()) {

```



```

int sno = rs.getInt("sno");
String sname = rs.getString("sname");
system.out.println("sno: " + sno + ", sname: " + sname);
}
stmt.close();
con.close();
} catch (ClassNotFoundException | SQLException e) {
} finally {
    scanner.close();
}
}
}
}

```

JDBC API:-

It Provides various methods and interfaces for easy communication with the database. It provides two Packages as follows, which contain the

Java SE and Java EE platform to exhibit
WORE (write once run anywhere) capabilities.
The java.sql package contains interfaces
and classes of JDBC API.