

7

3

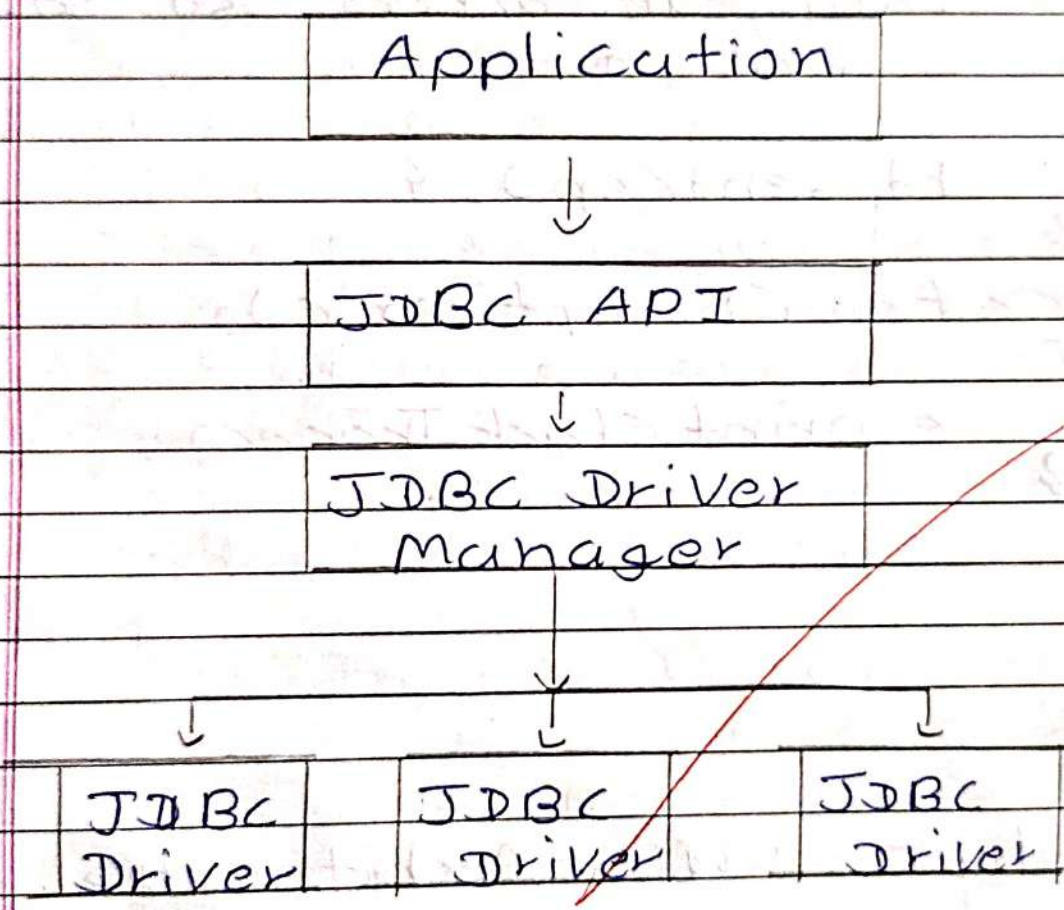
Explain JDBC Architecture.

JDBC is a Java API to connect with the database and execute any type of query.

સંયમઅને સાદગી દ્વારા જીવનમાં શાંતિ અને સંતોષ અનુભવાય છે.

JDBC Architecture contains Four part.

- i) Application
- ii) JDBC API
- iii) JDBC Driver Manager
- iv) JDBC Drivers.



i) Application:

It is a Java Applet, Java Swing or Java Servlet that communicate with data source.

Java Application is connected with Java API.

(ii) JDBC API :

JDBC API is used to Perform the SQL Query in the Java Program.

Using PreparedStatement Interface and CallableStatement Interface, we can perform any query in Java program.

(iii) JDBC Driver Manager:

JDBC Driver Manager Provides different Driver to connect with Database using Java Program.

JDBC Driver Manager contains Driver to communicate with Databases.

(iv) JDBC Driver :

JDBC Driver is communicate with database to respective data source.

There are Many type of JDBC Driver is used to communicate with database.

=> What is JDBC Driver? List the types of driver with explain type 4 driver.

JDBC Driver is communicate with database to respective data source.

JDBC Driver convert User Java Program request into the DBMS protocol.

There are Four type of JDBC Driver.

- 1) Type - 1 Driver
- 2) Type - 2 Driver
- 3) Type - 3 Driver
- 4) Type - 4 Driver

1 Type - 1 Driver :

Type - 1 Driver is called JDBC ODBC bridge driver.

This driver use ODBC driver to connect to the Database and convert JDBC Method into the ODBC Function calls.

This driver is easy to use and can be easily connected with any database.

## 2 Type - 2 Driver :

Type - 2 Driver is also know as a Native-API Driver.

This driver uses Native API driver which is use client-side Hdr libraries of the databases.

~~For~~ For use for Native API Driver we needs to be installed Native driver in each client machine.

## 3 Type - 3 Driver :

Type - 3 Driver is also know as Network Protocol Driver.

This driver Uses middleware that converts JDBC call into the vender-specific database protocol.

This Protocol does not required any type of client side library for database.

#### 4 Type-4 Driver:

Type-4 Driver is also know as Thin Driver.

The Thin Driver converts JDBC calls into vendor-specific database protocol.

Thin Driver gives better performance than all the driver and not required any client side or server side software.

⇒ Give the different way to create the JDBC connection with example.

For Create JDBC connection we have to use JDBC API.

Using JDBC API, we can create JDB database connection.

JDBC API contains different

interfaces to create JDBC Connection.

These are the basic JDBC API Interfaces.

(i) Driver Interface :

This Interface is used to implement JDBC Driver in Java Program to connect with Databases.

(ii) Connection Interface :

This Interface is used to create connection between Java application and databases.

(iii) PreparedStatement Interface :

PreparedStatement is used to execute parameterized query in Java application.

(iv) Callable Statement Interface :

Callable Statement is used to execute stored procedures and function in Java application.

## c) Result Set Interface

The Result set Interface is used to updating the data of a result set.

- This are the basic step for create JDBC Connection

- 1 Register Driver
- 2 Get Connection
- 3 Create Statamenent
- 4 Excute Query
- 5 Close Connection.

- Example:

```
import java.sql.*  
  
class Java  
{  
    public static void main (String  
        args []) throws Exception  
    {  
        Class.forName ("oracle.jdbc.  
            driver.OracleDriver");  
  
        Connection con = DriverManager  
            .getConnection ("jdbc:Oracle:  
                thin:@localhost:1521:xe",  
                    "root", "root");  
    }  
}
```



```
- CallableStatement stm = con.  
prepareCall("call insert  
R(?,?)");
```

```
stm.setInt(1, 10110);
```

```
stm.setString(2, "K");
```

```
stm.execute();
```

```
}
```

```
}
```

\* Write a Steps to connect to the Database.

=> This are the Basic steps to connect the database.

1 Register the Driver Class:

Using this step, we have to register driver class in java application.

Using `forName()` Method, we can register the Driver class.

Syntax :

```
Class.forName("Driver Class");  
Name
```

2 Create the connection Object:

Using `getConnection()` Method, we can establish connection between database and driver class.

Syntax:

Connection Connection =  
Object

DriverManager.getConnection(" ");

3 Create the Statement Object:

The createStatement() method of Connection interface is used to create statement.

The object of statement is use to execute queries with the database.

Syntax:

Statement Statement =  
Object

con.createStatement();

4 Execute the Query:

The executeQuery() method of Statement is used to execute queries to the database.

Syntax :

```
ResultSet ResultSet =  
    Object
```

```
stmt.executeQuery("Query");
```

5 Close the connection Object:

The close() method of connection is used to close the connection.

Syntax :

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