Date
Page

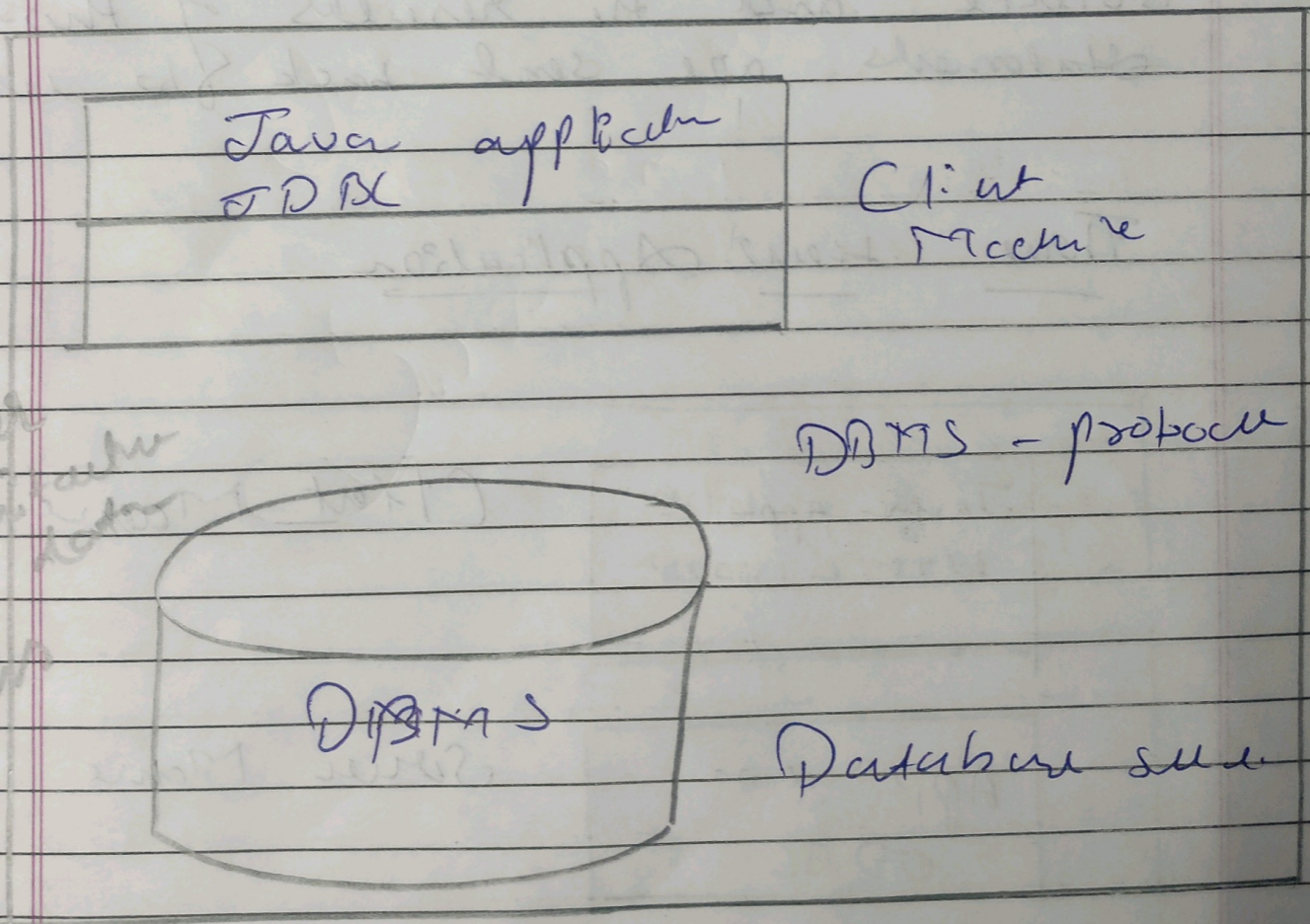
Name: Ajinkya Rathod

DAD

Explain JDBC Architecture

Ans. The JDBC API support both two tier and three tier processing models for data base access.

-> a Tier Architecture

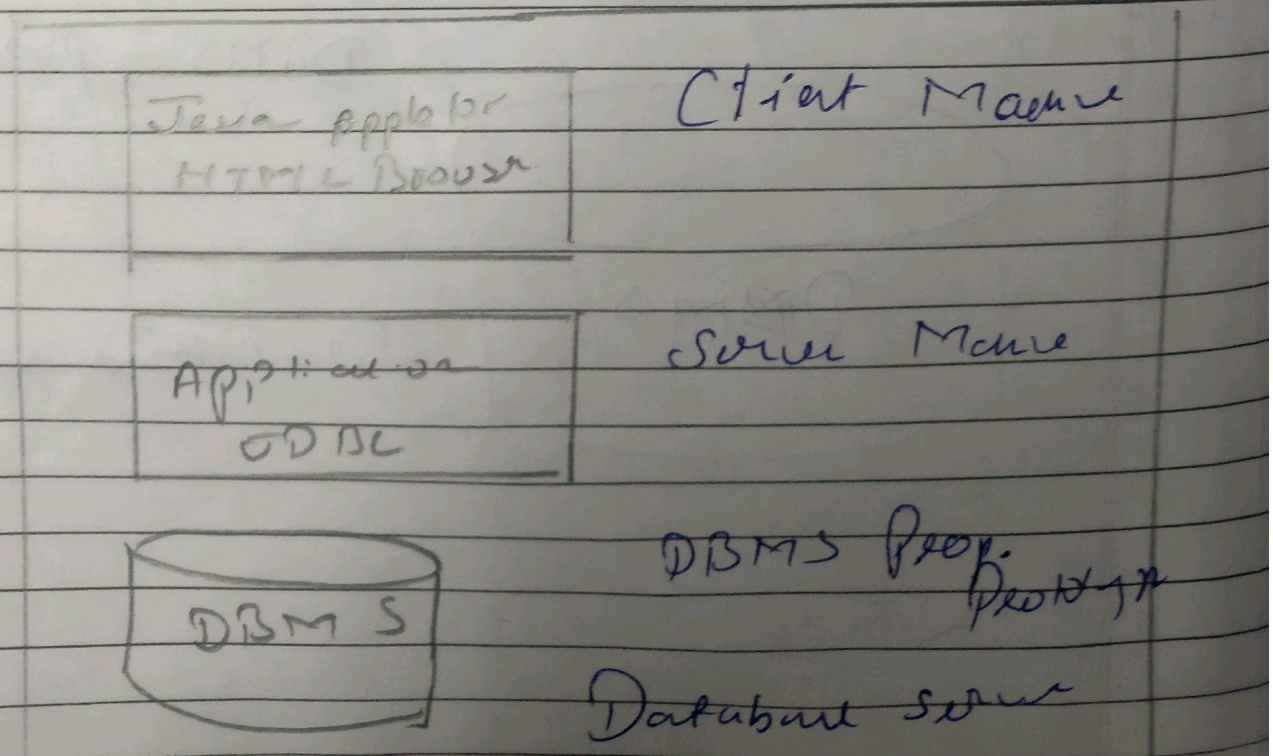


In a tier model, Java application talks directly to data source.

→ This requires a JDBC driver that can communicate with particular data source being accessed.

→ A user's commands are delivered to the database or other data source and the results of those statements are sent back to user.

Three tier Application



-> In this, commands are sent to "middle tier" of services, which then sends the commands to data source.

-> The data source process the commands and send the result back to the middle tier then send them to server.

Q2 JDBC Driver Types

* Type-1 (JDBC-ODBC Driver)

-> This does not directly interact with database. This relies on ODBC driver to communicate with database.

-> It depends on ODBC driver.

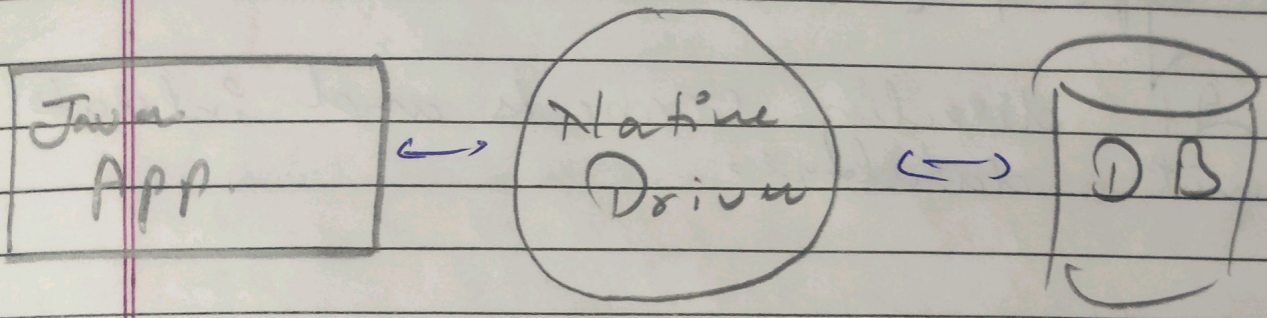
-> The implementation of this driver makes use of native methods, in order to make standard ODBC calls.

* Type-2 driver

It is partially written in Java and partially in native code.

-> It is not dependent on any other driver or application like ODBC.

Teacher's Signature.....



* Type-3 (Net Driver)

→ These drive interacts directly with database via some network based middleware.

→ The implementation of this type of JDBC driver with software based architecture doesn't use of native code.

PAGE NO.
DATE :

* Type-4 (Pure Java Driver)

→ It directly connects and interacts with database.

→ Not dependent on any driver.

Q. Steps of registering Driver class.

1. Load driver class into Java program

Ex: Class.forName("com.mysql.jdbc.Driver");

2. You can set driver property

java -jdbc.drivers=com.mysql.jdbc.Driver program.name

- or your app can set system property with call such as.

System.setProperty("jdbc.drivers", "com.mysql.jdbc.Driver");

Teacher's Signature.....

Q. Discuss the following

Execute Update()

→ This method is used to execution of DML statement.
(INSERT, UPDATE, DELETE)

→ This method return int value, count of affected rows.

Q.

Q. Execute Query

This method is used to retrieve data from database using SELECT query.

→ This method returns the ResultSet Object that returns the data according to the Query

This method is used to execute query select query.

(3) Execute

This statement used for all types of SQL statement

→ Return Boolean value of TRUE or FALSE.

→ If true, returns ResultSET obj else int value

Q11 SQL Exceptions 5 types

(1) SQL Warning

- An exception that provides information on database access warnings.
- Warnings may be retrieved from Connection and ResultSet objects.

public SQLWarning(String reason, String SQLState)

(2) Batch Update Exception

The subclass of SQLException thrown when an error occurs during a batch update.

In addition to info provided by SQLException, a BatchUpdate provides an update counts for all commands that were executed successfully during batch update,

PAGE NO.
DATE :

(3) Row Set Warning

An instance of SQL Exception that provides information about database access errors or warnings.

(4) Serial Exception

Indicates an error in serialization or de-serialization of SQL types such as BLOB, CLOB, STRUCT or ARRAY in addition to SQL types such as DATA and JAVA OBJECT

(5) SQL Client Info Exception

The subclass of SQL Exception is thrown when one or more client info properties could not be set on a connection.

This provides a list of client info properties that were not set.

Teacher's Signature.....

PAGE NO.
DATE :

XML

Q1 DTD.

→ To specify document structure, you can supply a DTD or XML schema definition

→ A DTD or schema contains rules that explain how document should be formed.

Ex DTD might contain a rule:

<!ELEMENT book (name, size)

<?xml version="1.0">

<!DOCTYPE config [

<!Element config ...)

more rules

])

<config>...

</config>

Teacher's Signature.....

Q. DTD and XML

1. DTD are declaration that define a document type of XML

XSD does this in XML DOC.

2. NO namespaces

namespace are supported

3. No datatype

→ Support datatypes

4. Not extensible

→ Extensible

Q. DOM vs SAX

	DOM	SAX
→	DOM parse is tree based parser.	SAX is event based
→	DOM loads whole XML docunt in memory.	SAX loads a small part of XML in memory.
→	Faster than XML col. more memory.	Not faster than DOM
→	Good for large file	Best for small size.

PAGE NO.
DATE :

Q. STAX

→ STAX is stream / event oriented XML parser.

→ The model over here is your handle does not calls the parser, not the other way around.

→ Thus your handle class controls when parser is to move on to next event in input.

→ In other words your handle "pulls" the XML events out of parser.

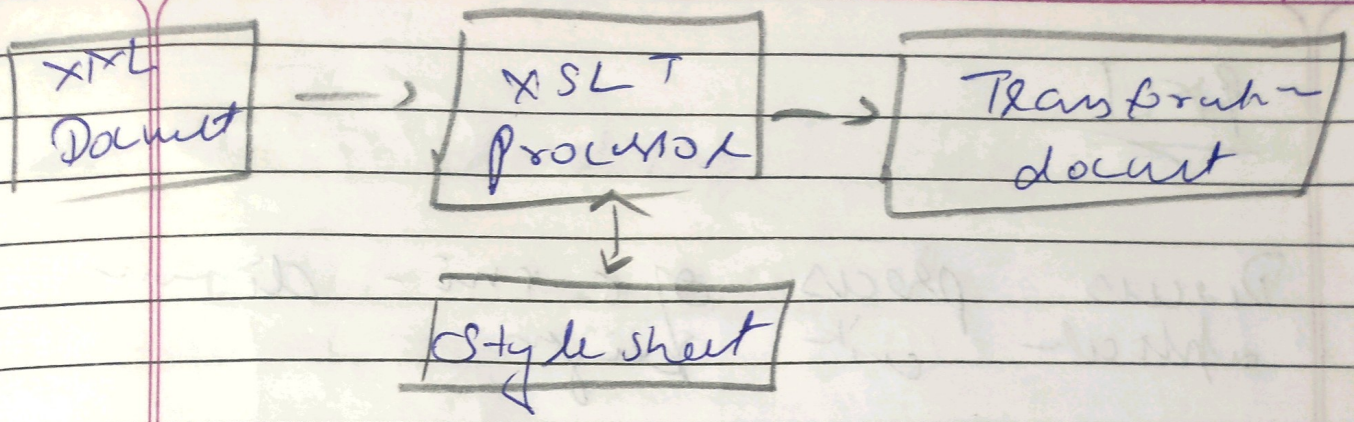
(3) XSLT or XSL

→ XSLT is Extensible stylesheet language Transformations. (XSLT) API.

→ It is not easy to write a DOM tree to an output stream. The exact ahead is to use XSLT API.

→ You need to provide an XSLT stylesheet that describes the conversion of XML of source into some other format.

→ The XSLT processor reads an XML document and stylesheet and produces the desired output.



(4) DTD

Refer Q1.

(5) Schema

The specify the document structure for can specify XML schema definition

→ A schema contains rules that explain how a document should be formed by specifying the legal child and attributes for each elements

PAGE NO.
DATE :

Q. RMI

Discuss process of a rmi - distributed application with diagram

→ RMI application are often comprise to two program, server program

→ A typical server application creates a no. of remote objects, makes references to those remote objects accessible and waits for client to invoke methods on those remote objects.

→ RMI provides the mechanism by which the server and client communicate and pass all forth such as application & security reference to as a distributed object application..

⇒ Distributed Object App. Needs

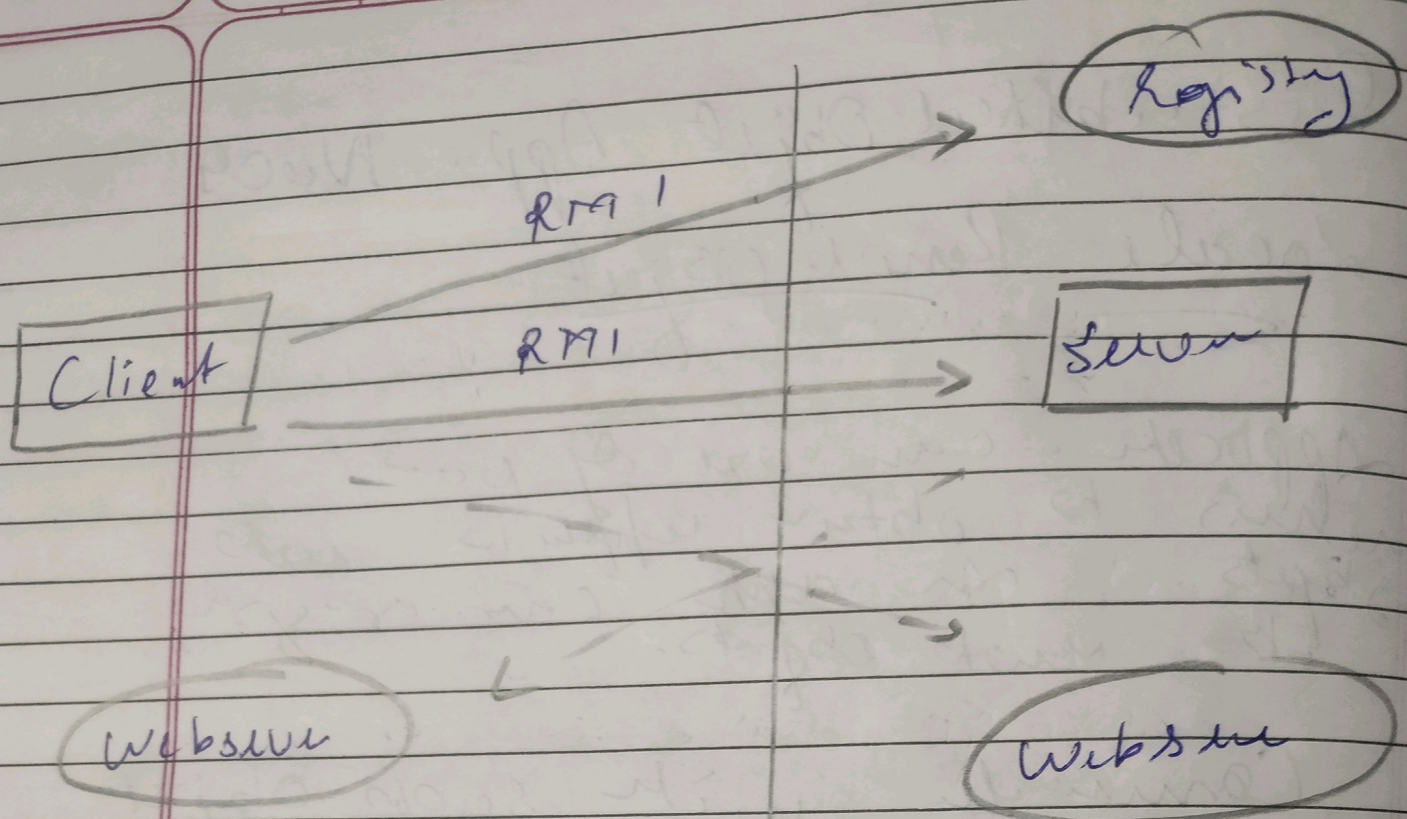
* Locate Remote Object

Applicat. can use one of two methods to obtain refs to remote objects. An app can register its remote objects.

* Communicate with remote object

Details of comm. between remote objects are handled by RMI. To the programmer, remote method invocation.

* Load class byte codes for objects that are passed as parameters or return values.



The illustration... an RMI distributed application... registry to obtain... a remote object...

PAGE NO.
DATE :

Q. Stub and skeleton

Ans Stub for a remote object acts as proxy for remote object.

= In RMI, a remote object is represented by a remote object interface.

- when stubs is invoked
- marshalled params
- waits for result
- returns value to caller func
- finds deserialization.

Teacher's Signature.....

DATE:

Skeleton

→ Responsible for dispatching the call to actual remote object implementation

→ Unmarshals incoming function

→ Invokes on remote obj.

→ marshals results to caller

Q2

```
import java.rmi.*;
import java.rmi.server.*;

class AdderRemote extends UnicastRemoteObject
    implements Adder
{
    AdderRemote() throws RemoteException
    {
        super();
    }
    public int add(int x, int y)
    {
        return x+y;
    }
}
```

Step-4 Start the RMI Registry

5 - Create and execute the server application program

6 - import java.rmi.*;

import java.rmi.server.*;

public class MyServer {

public static void main(String args[]) {

try {

Adder stub = (Adder) Naming.lookup("rmi://localhost:5000/sonoo", stub);

}

}

6 = Create and execute client app. program.

Internationalization

- Collation

→ Application that sort through n text perform frequent String comparisons.

→ If your application audience is limited to people who speak English ; you can probably perform String Comparison using the String. compare to method

* Normalization

It is the process by which you can perform certain transformations of text to make it reconsiche for a way which it may not have been before.

→ The Unicode standard defines 4 forms

- → D
- → KD
- → C
- → KC

(for string)

Ch 2 Message Formatting

The Java library has message Format class that formats text with variable parts.

→ Message Format takes a set of objects, formats them ~~Fmt~~ and then inserts the formatted string into pattern at appropriate places.