

6. Interacting with database

JDBC is the Specification from Sun microsystem that provides standard abstraction API for Java application to Communicate with different Databases.

JDBC → Java Database Connectivity

ODBC → open database Connectivity

Role of JDBC Driver is to process sql request & generate results according to query

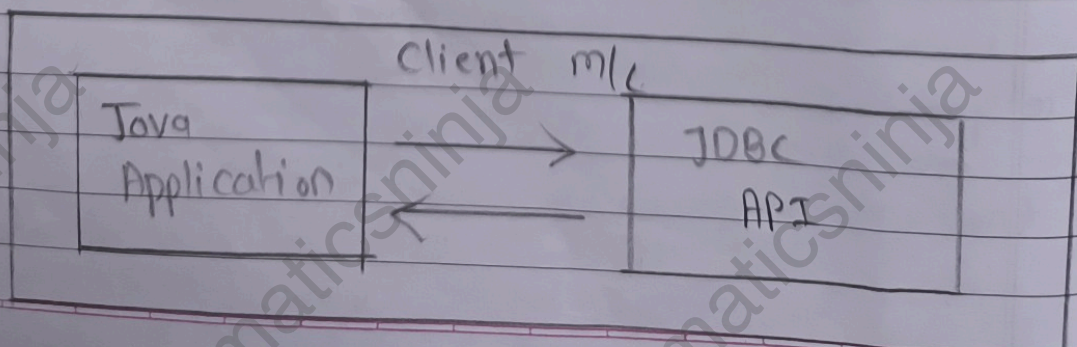
• Some classes & Interfaces

- 1] DriverManager Statement
- 2] Connection Prepared Statement
- 3] Result Set

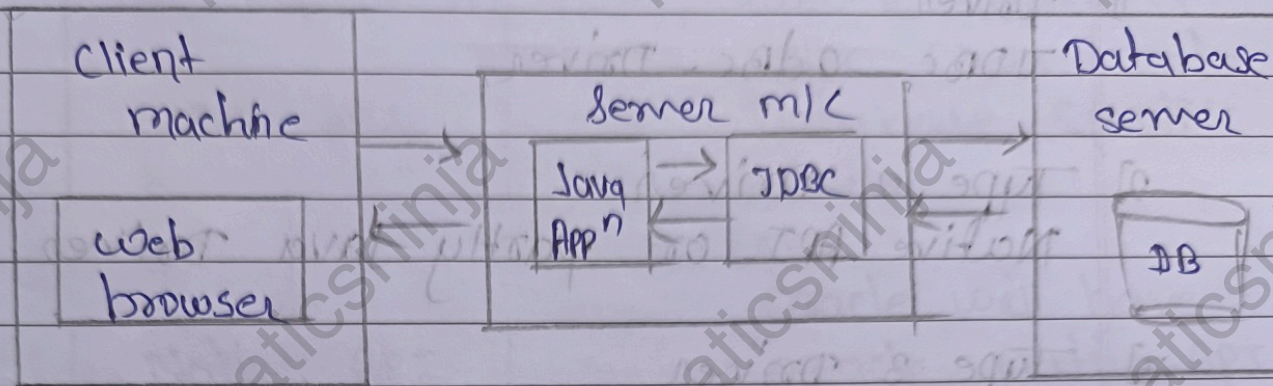
• There are two types of JDBC Architecture

- 1] 2 tier Application
- 2] 3 tier Application

1] 2 tier Application



2] 3 tier Application



⇒ 2 tier

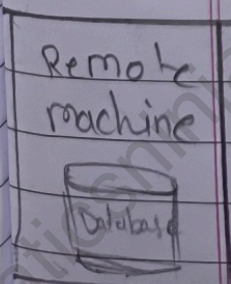
Two tier Architecture provides Direct Communication between Java Application with Database

It requires JDBC Driver that can help to communicate with particular Database

⇒ 3 tier

In 3 tier model commands are sent to HTML browser to middle services that is Java application which can send command to particular Database

The middle tier is written in C / C++



Types of JDBC Driver

1] Driver

JDBC - ODBC - Driver

2] Type 2 Driver

Native API or partly Java Drives

3] Type 3 Driver

All Java / Net protocol Driver

4] Type 4 Driver

Native protocol / All Java driver

5] JDBC - ODBC - Bridge Driver

Java Application

JDBC
API



Bridge
Driver



DAI

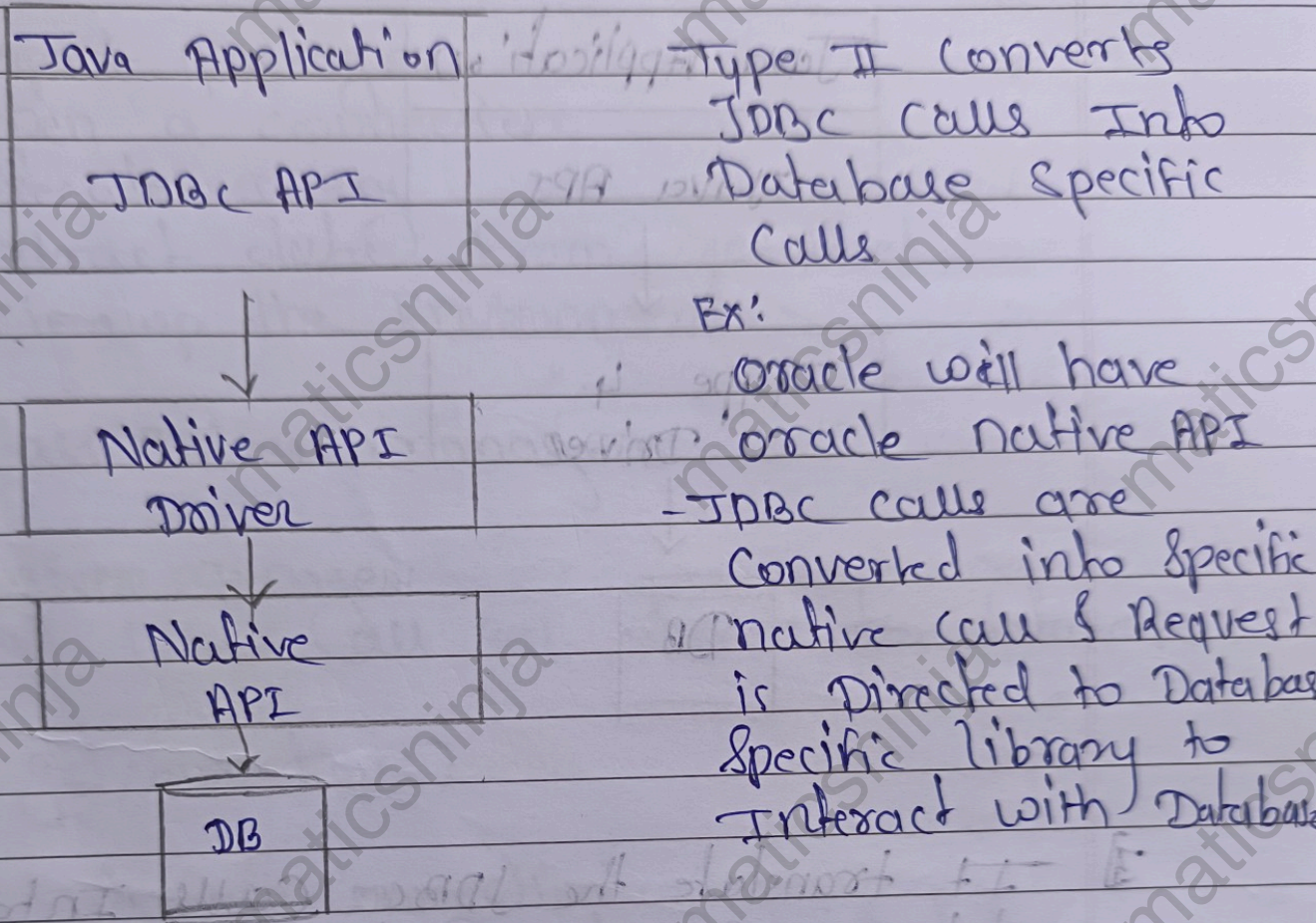
Driver

ODBC

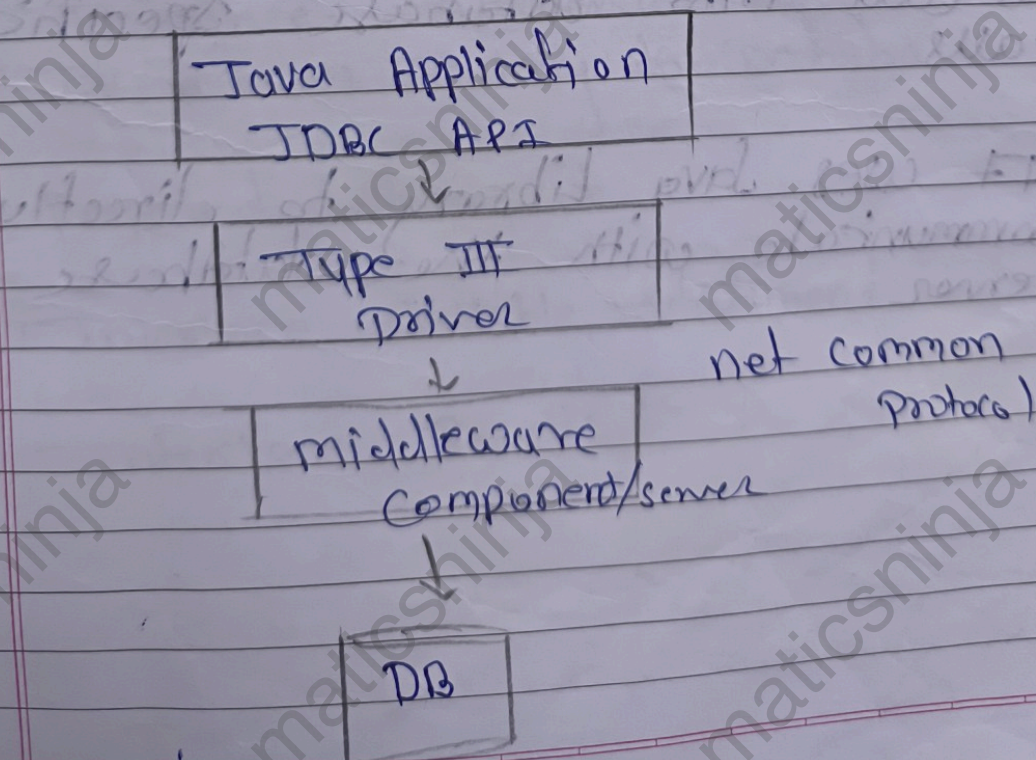
DB

It translate all JDBC
calls into ODBC calls
and sends them into
ODBC Driver

2) Native API / Partly Java Driver



3) All Java / Net protocol Driver



4) Native protocol / All Java Driver

Java Application

Java API

Type 4
Driver

DB

3 It translate the JDBC calls into Database Server Independent after that middleware server convert this call into database specific calls

4 It uses Java Library to directly communicate with the database server

⇒ 4 Steps of JDBC Application

- 1] import packages
- 2] open a connection
- 3] Execute query
- 4] extract data From resultset
- 5] Cleanup the Environment.

⇒ Class Driver manager

1] Driver manager

— It loads all sql drivers to database
getconnection()

2] Interface Connection

— It makes Connection to Database

3] Interface Statement

— It makes query to Database

4] Prepared Statement

5] Interface Callable Statement

6] Interface Resultset

- Class Class
This class is used to load JDBC driver
for connecting Database

forName ()

Ex:

```
import java. sql.* ;  
public class CreateDBTable
```

```
{  
    Connection con; DriverManager getConnection(  
    CreateDBTable ( ) "jdbc:")
```

```
{  
    try  
    {
```

```
        Class.forName ("Oracle. jdbc. Driver oracleDriver
```

```
        // create Connection object
```

```
        con = DriverManager. getConnection ("jdbc.  
        oracle: @localhost: 1521: xe", "local", "test")  
        // create prepared statement object.
```

```
        Preparedstatement ps = con. prepareStatement  
        (" Create table Student (Stelid int  
        primary key, stuName Varchar (20) not null")  
        // execute sql query
```

```
        int Count = ps. executeUpdate();  
        System. out. println (Count);
```



```

System.out.println (" Table Created  

Successfully");
}
catch (exception e)
{
e.printStackTrace();
}
public static void main (String args[])
{
new CreateDBTable();
}

```