**Q #1) What are the components of J2EE applications?**

**J2EE**

* Client-tier components. Run on the client machine.
* Web tier components. Run on the J2EE server.
* Business tier components. Run on the J2EE server.
* Enterprise information system software (EIS software).Runs on the **EIS** server.

**Q #2) What are the J2EE client types?**

* Applets
* Application clients
* Java Web Start enabled clients, by **Java Web Start technology**.
* Wireless clients, based on **Mobile Information Device Profile (MIDP)** technology.

**Q #3) What is a J2EE container?**

**Ans)** The interface between a component and the low-level platform with specific functionality that support the component is called a **container**. Application server maintains control and provides relevant services through an interface or framework calls as a **container**.

**Q #4) What are defined as web components?**

**Ans)** **Java Servlets** and Java server pages technology components are identified as web components. Servlets dynamically receive requests and make responses. JSP server pages also behave as **Java Servlets** but facilitate to create more static contents.

**Q #5) Describe the MVC on struts?**

**Ans)** **MVC** stands for Model View Controller. Each section in Model View Controller can describe as follows.

* **Model –**Model represents the internal state of the system as a set of single or many Java Beans.
* **View –** Most often view is a constructed using **Java Server Pages (JSP)** technology.
* **Controller –**The controller is the process that focuses on client request receiving and produce next level of a user interface according to the request. The primary component of the controller in the framework is **“ActionServlet”** servlet class.

**Q #6) Define JSF?**

**Ans)** **JSF** stands for Java Server Faces. It is the user interface (UI) designing framework for Java Web Applications developments. There are set of reusable UI components associated with JSF. Also, JSF based on Model-View-Controller (MVC) design concepts and patterns. The automated data saving process from form to server and display at client side is also handling by JSF.

**Q #7) What is Hashtable?**

**Ans)** Hash table is a Collection Synchronized objects. It allows Null values but not duplicate values. Hash table is like a HashMap.

**Q #8) Define Hibernate?**

**Ans)** Hibernate is an open source object-relational mapping and query service which facilitate to write **Hibernate Query Language (HQL)** scripts instead of **Structured Query Language (SQL)**scripts. It is faster and easy than writing native **SQL**. Hibernate has more powerful object oriented contents like associations, inheritance, and polymorphism. Also, Hibernate has a powerful composition and collections. Hibernate allows making queries using Java base approach.

**Q #9) What are the identified limitation of hibernate?**

**Ans)**

* **Slower in action –** In execution of HQL queries take more time than it executes directly.
* Only composite keys support available and it prevents advanced query options.
* No shared value type references available.

**Q #10) What the identified advantages are of hibernate?**

**Ans)**

* Database and vendor independence application.
* Standard Object-relational mapping support.
* Domain object mapping for a relational database.
* Better performance than Java Database Connectivity.
* Java Persistence **API**based applications.

**Q #11) Describe ORM?.**

**Ans) Object-Relational mapping (ORM)** can describe as follows.

The mapped objects in a Java class to the tables of the relational database using metadata which describes the database and object mapping. The working method is to transform data from one representation to another.

**Q #12) What are the advantages of Object-Relational mapping (ORM)?**

**Ans)**

* **Productivity –** Reduce the time for data access coding with help of automatic code creation base on the defined data model.
* **Maintainability –** All code generated from **ORM** are well tested. Only the developer need to create the correct functionality
* **Performance –** The code generated from **ORM** completely manages the data access needs of the application. No need to create any data access code to create and also the code is optimized for speed up the data access process.
* **Vendor independence –** The code generated from **ORM** is not depending on the vendor. This is to increase the portability of the application.

**Q #13) What is the use of method save()?**

**Ans)** In hibernate this method is used to stores an object into the database. There is a check for duplicate records before inserting it.

**Q #14) What is the use of method saveorupdate()?**

**Ans)** In hibernate this method is used to update an object using identifier. When the value for the identifier is NULL then the method direct to call save().

**Q #15) What is the difference between load() and get()?**

**Ans)** When the object not available in either cache or database, load() thrown an exception. No null return from load().

When the object not available in either cache or database, get() returns null.

**Q #16) What is mean by connection pooling?**

**Ans)** Simply connection pooling is a mechanism to re-use the existing connections. The **pooling**mechanism maintains a number of already created object connections and when there is a demand the mechanism directly use existing connection without creating a new one.

**Q #17) Define the Collection types in Hibernate?**

**Ans)** One-to-many reference is defined as a collection. There are five main collection types associated with J2EE.They are

Set type, List type, Array type, Map type, Bag type

**Q #18) Define thin client?**

**Ans)** A program interface that does not have any operations like database queries, complex business rules or any connection to the third-party application is called a thin client.

**Q #19) Describe the file types \*.ear, \* .jar and \*.war?**

**Ans)**

* **\*.jar files –** Property file contains libraries, resources and accessories are included with the \*.jar file extension.
* **\*.war files –** The files that need to development of web application (HTML, java scripts, JSP) included with a \*.war file extension.
* **\*.ear files –** The files for Enterprise Java Beans modules for the application is save as \*.ear files.

**Q #20) What spring is in related to J2EE?**

**Ans)** **Spring** is an open source application that reduces the complexity of enterprise application development. Spring is based on an inversion of control or dependency injection design patterns.

**Q #21) What are the advantages in the use of spring for application development?**

**Ans)**

* **Plain Old Java Object (POJO)** based development facilitates to re-use existing components.
* Possible to reduce development cost by improving the productivity of the application development.
* Improve the testability of application with dependency injection.
* Improve maintainability with reduce code coupling.
* No need to have an application server and works on enterprise service.

**Q #22) Discuss the benefit of Spring Framework?**

**Ans)**

* Possibility to organize middle tier objects in an efficient way.
* Easy initialization for properties.
* Easily testable components.
* Lightweight container.
* Possibilities to use configure management service of spring in any runtime environment with whatever architectural layer.

**Q #23) Describe servlet?**

**Ans)** Server side component that provides a powerful mechanism to create server side programs is called a **servlet**. There are servlets available with a design for various protocols. Servlet is also server and platform independent. Most commonly use protocol for the servlet is **Hypertext Transfer Protocol (HTTP)**.Also, a servlet is a pure java object.

**Q #24) Describe the phases of servlet lifecycle?**

**Ans)**

* **Class loading phase –** Web container loads the servlet class file (\*.class).
* **Instantiation phase –** By calling default no-arg constructor the servlet class gets Instantiated.
* **Initialize phase –** The method Init () called in this phase in only one time of the lifetime of a servlet. Servlet configuration is assigned to the servlet.
* **Request Handling phase –** In this phase, only servlets spends most of the time. Servlet provides the services to various requests by calling Service ().
* **Removal phase –** The destroy () function calls before servlet destruction. Garbage collection occurs later.

**Q  #25) What are the difference types of JSP tags?**

**Ans)** ***There are 4 different types of tags associated with JSP.They are mentioned below***

* Directives
* Declarations
* Scriplets
* Expressions

**Q #26) Describe action form?**

**Ans)** A java bean that is associated with single or multiple action mapping is called action form. Action form objects are automatically populated at server end when data has been enter from the client from a**user interface (UI)**.

Session states of a web application are maintained by action forms.

**Q #27) Describe Secure Socket Layer (SSL)?**

**Ans)** The technology that use to communicate between the web server and the web browser is called **Secure Socket Layer (SSL)**.More especially SSL is a protocol that describes how algorithms to be used in encryption. The technology established an encrypted link between two parties and this link is allowed to secure transmission of sensitive information such as login credentials, credit/debit card information and social security numbers.

**Q #28) What id URL?**

**Ans)** **URL** stands for Uniform Resource Locator and it is the textual reference writing standard to an arbitrary piece of data in the **World Wide Web (WWW)**.The general structure of URL is as follows.

***protocol://host/local info***

* ***protocol* ->** Protocol for fetching the object ( eg – HTTP, FTP)
* ***host* ->** Internet name of the targeted host.
* ***local info* ->** String passed to the protocol handler on the remote host. In many cases, it is a file name with extension.

**Q #29) What is URN?**

**Ans)** **URN** stands for Uniform resource name. It is the unique identifier that identified an entity. But the information of where the entity is located is not available.

**Q #30) What are the phases of a servlet life cycle?**

**Ans)** ***Following steps are associated with servlet life cycle.***

* Loading of the servlet class.
* Instantiation of Servlet.
* Execution of init method.
* Request handling phase. In this phase service methods will call.
* Removal from service phase. In this phase destroy method will call.

**Q #31) Is that Servlet is pure java object or not?**

**Ans)** Yes, Servlet is pure java object.

**Q #32) What is EJB?**

**Ans)** **EJB** stands for Enterprise Java Beans. It is the server side components that executes in EJB container and encapsulates the business logic for the enterprise application.

**Q #33) What are the system services of EJB container?**

**Ans)** ***EJB Container provides following system services.***

* Persistence
* Security
* Transaction
* Connection pooling
* Component lifecycle management
* Threading

**Q #34) What are the design principles for EJB?**

**Ans)**

* Behavior of the **EJB** application is specified by interfaces
* EJB applications are loosely coupled and tired.
* Implementation is hidden from the client side.
* The **EJB** container supports the application developer.
* The API to the application is in session tier.
* The API to the data sources is in entity tier.

**Q #35) What are the advantages of EJB components?**

**Ans)**

* There is a support for the integration of components from different vendors.
* Possibility authored **EJB** components detailed knowledge of the environment.
* Possibility to assembled applications from separate components source.
* Interaction with its clients is entirely specified in terms of java interfaces.
* Portability support.
* Does not maintain resources.

**Q #36) What are the Basic and subtypes of Enterprise Java Beans (EJB)?**

**Ans)** ***Two main types and subtypes of EJB are as follows.***

* **Session Beans**
  + Stateful session beans
  + Stateless session beans
* **Entity Beans**
  + Bean Managed Persistence (BMP)
  + Container Managed Persistence (CMP)
  + Message Driven Beans

**Q #37) What is the description for expression element?**

**Ans)** The expressions that use to write dynamic content back to the client browser are called **expression elements**.

**Q #38) What are the two types of comments supported by JSP?**

**Ans)** ***There are two types of comments are supported by JSP.***

* HTML comment.

[HTML comment](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/HTML-comment.jpg)

* JSP comment.

[JSP comment](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2017/03/JSP-comment.jpg)

**Q #39) What is called JSP directive?**

**Ans)** **JSP** directive is the mechanism to provide Metadata information to web container about JSP file. In the translation and compilation phases of the **JSP** life cycle, these Metadata use by the web container.

**Q #40) What are the different types of JSP directive?**

**Ans)** ***There are 3 different types of JSP directives available.***

* Page directive
* Include directive
* Taglib directive

**Explain what do you mean by JDBC?**

Short for Java Database Connectivity, a Java API that enables Java programs to execute SQL statements. This allows Java rograms to interact with any SQL-compliant database. Since nearly all relational database management systems (DBMSs) support SQL, and because Java itself runs on most platforms, JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMSs. JDBC is similar to ODBC, but is designed specifically for Java programs, whereas ODBC is language-independent

**What are the components that constitute JDBC ?**

JDBC Components include connection Pools, Data Sources, and MultiPools

**How to you load the drivers in JDBC?**

Class.forName() method is used in JDBC to load the JDBC drivers dynamically

**What do you mean by batch updates ?**

if you want to execute a set of statements, i.e. SQL statements at a time then we use batch update statement.  
resultset=pst.batchUpdate();

**Explain how you can establish a connection ?**

Loading Drivers  
Class.forName(“Driver”);  
Getting connection  
Connection con = DriverManager.getConnection(url,”myLogin”,

“myPassword”);

**What are the  different types of statements in JDBC ?**

java.sql.Statement – Top most interface which provides basic methods useful for executing SELECT, INSERT, UPDATE and DELETE SQL statements. java.sql.PreparedStatement – An enhanced verion of java.sql.Statement which allows precompiled queries with parameters. It is more efficient to use java.sql.PreparedStatement if you have to specify parameters to your SQL queries.  
java.sql.CallableStatement – Allows you to execute stored procedures within a RDBMS which supports stored procedures (MySQL doesn’t support stored procedures at the moment).

**Have you used prepared statements? Where have you used prepared statements ?**

Yes. If you want to execute a Statement object many times, it will normally reduce execution time to use a PreparedStatement object instead. The main feature of a PreparedStatement object is that, unlike a Statement object, it is given an SQL statement when it is created. The advantage to this is that in most cases, this SQL statement will be sent to the DBMS right away, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement ‘s SQL statement without having to compile it first. Although PreparedStatement objects can be used for SQL statements with no parameters, you will probably use them most often for SQL statements that take parameters. The advantage of using SQL statements that take parameters is that you can use the same statement and supply it with different values each time you execute it.

**How do you create JDBC statements ?**

Connection con = null;  
Statement st = null;  
// Obtain connection here  
st = con.createStatement();  
ResultSet rs = null;  
rs = st.executeQuery(“SELECT \* FROM users”);  
int recordsUpdated;

recordsUpdated = st.executeUpdate(“DELETE FROM users WHERE user\_id =

1”);

**How do you retrieve data from a result set, explain with an examplet?**

Example:  
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery(SELECT COF\_NAME, PRICE FROM

COFFEES\_T);  
while (rs .next() )  
{  
//Iam assuming there are 3 columns in the table.  
System.out.println ( rs.getString(1));  
System.out.println(rs.getString(2));  
System.out.println(rs.getString(3));  
}  
//Never forget to close the resultset, statement & connection  
rs.close(); //First  
stmt.close(); //Second  
con.close(); //Last  
System.out.println(“You are done”);

**Explain about stored procedure ?**

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. Stored procedures are used to encapsulate a set of operations or queries to execute on a database server. For example, operations on an employee database (hire, fire, promote, lookup) could be coded as stored procedures executed by application code. Stored procedures can be compiled and executed with different parameters and results, and they may have any combination of input, output, and nput/output parameters.

**What are the different tasks of JDBC ?**

**Following are the tasks of JDBC**  
Load the JDBC drivers  
Register the drivers  
Specify a database  
Open a connection to database  
Submit a query to database  
Gets the results

**When do we look for  batch updates ?**

Let’s say there are 100 records need to be insert. If we execute normal statemets the no of transactions will be 100 (in terms of connection making to DB). using batch updates we can add 100 rec to batch and the no of transactions will be only one in this case. This will reduce the burdon on db, which is very costly in terms of resources.

**Explain the way in which you can invoke a stored procedure from java ?**

You can call a stored procedure using Callable statements  
CallableStatement cs = con.prepareCall(“{call StoredProc}”);  
ResultSet rs = cs.executeQuery();  
What packages are being used by JDBC  
Following packages are used in JDBC  
java.sql  
javax.sql

**How can you get the resultset of Stored procedure ?**

CallableStatement cstmt;  
ResultSet rs;  
int i;  
String s;  
…  
cstmt.execute();// Call the stored procedure 1  
rs = cstmt.getResultSet();// Get the first result set 2  
while (rs.next()) {               // Position the cursor 3  
i = rs.getInt(1);          // Retrieve current result set value  
System.out.println(“Value from first result set = ” + i);   // Print the

value  
}  
cstmt.getMoreResults();  // Point to the second result set 4a  
// and close the first result set  
rs = cstmt.getResultSet(); // Get the second result set 4b

while (rs.next()) {               // Position the cursor 4c  
s = rs.getString(1);   // Retrieve current result set value  
System.out.println(“Value from second result set = ” + s);  
// Print the value  
}  
rs.close();                          // Close the result set  
cstmt.close();                    // Close the statement

**When do we set setAutoCommit(false) ?**

The DML operations by default are committed. If we wish to avoid the commit by default, setAutoCommit(false) has to be called on the Connection object.Once the statements are executed, commit() has to be called on the Connection object explicitly

**What is a data source ?**

A DataSource class brings another level of abstraction than directly using a connection object. Data source can be referenced by JNDI. Data Source may point to RDBMS, file System , any DBMS etc

**Explain the difference between Resultset and Rowset ?**

**RowSet**

The interface that adds support to the JDBC API for the JavaBeansTM component model. A rowset, which can be used as a JavaBeans component in a visual Bean development environment, can be created and configured at design time and executed at run time. The RowSet interface provides a set of JavaBeans properties that allow a RowSet instance to be configured to connect to a JDBC data source and read some data from the data source. A group of setter methods (setInt, setBytes, setString, and so on) provide a way to pass input parameters to a rowset’s command property. This command is the SQL query the rowset uses when it gets its data from a relational database, which is generally the case. The RowSet interface supports JavaBeans events, allowing other components in an application to be notified when an event occurs on a rowset, such as a change in its value. The RowSet interface is unique in that it is intended to be implemented using the rest of the JDBC API. In other words, a RowSet implementation is a layer of software that executes “on top” of a JDBC driver. Implementations of the RowSet interface can be provided by anyone, including JDBC driver vendors who want to provide a RowSet implementation as part of their JDBC products. A RowSet object may make a connection with a data source and maintain that connection throughout its life cycle, in which case it is called a connected rowset. A rowset may also make a connection with a data source, get data from it, and then close the connection. Such a rowset is called a disconnected rowset. A disconnected rowset may make changes to its data while it is disconnected and then send the changes back to the original source of the data, but it must reestablish a connection to do so.

**ResultSet**

A table of data representing a database result set, which is usually generated by executing a statement that queries the database. A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set. A default ResultSet object is not updatable and has a cursor that moves forward only. Thus, you can iterate through it only once and only from the first row to the last row. It is possible to produce ResultSet objects that are scrollable and/or updatable. The following code fragment, in which con is a valid Connection object, illustrates how to make a result set that is scrollable and insensitive to updates by others, and that is updatable. See ResultSet fields for other options.

**How do we retrieve warning ?**

SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do. They simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object.

E.g.

SQLWarning warning = stmt.getWarnings();

if (warning != null) {

while (warning != null) {

System.out.println(“Message: ” + warning.getMessage());  
System.out.println(“SQLState: ” + warning.getSQLState());  
System.out.print(“Vendor error code: “);  
System.out.println(warning.getErrorCode());  
warning = warning.getNextWarning();

}

}

**How many statements can we create with one connection ?**

There is no such limit on number of statements to be created

**How to make updated to the  Updatable Result Sets ?**

Usinf JDBC 2.0 API we have the ability to update rows in a result set. For this we need to create a ResultSet object that is updatable.  
For this, we pass the ResultSet constant CONCUR\_UPDATABLE to the createStatement method.  
Connection con =DriverManager.getConnection(url, “myLogin”, “myPassword”);  
Statement stmt =con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE);  
ResultSet uprs =stmt.executeQuery(“SELECT NAME, SALARY FROM EMPLOYEES”);

**How can you move the cursor in scrollable result sets**

In JDBC 2.0 API we have the ability to move a result setï¿½s cursor backward as well as forward.  
We can also move the cursor to a particular row and check the position of the cursor.  
Statement stmt = con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_READ\_ONLY);  
ResultSet srs = stmt.executeQuery(‘SELECT NAME, SALARY FROM EMPLOYEES’);  
The first argument is one of three constants added to the ResultSet API to indicate the type of a ResultSet object: TYPE\_FORWARD\_ONLY, TYPE\_SCROLL\_INSENSITIVE , and TYPE\_SCROLL\_SENSITIVE. The second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable: CONCUR\_READ\_ONLY and CONCUR\_UPDATABLE. Make sure that when you specify a type, you must also specify whether it is read-only or updatable. Specifying the constant TYPE\_FORWARD\_ONLY creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a ResultSet object, you will automatically get one that is TYPE\_FORWARD\_ONLY and CONCUR\_READ\_ONLY.

**What is the query used to display all tables names in SQL Server (Query analyzer)?**

select \* from information\_schema.tables

**How many types of JDBC Drivers are present and what are they?**

– There are 4 types of JDBC Drivers

JDBC-ODBC Bridge Driver  
Native API Partly Java Driver  
Network protocol Driver  
JDBC Net pure Java Driver

**I made my class Cloneable buts still I am not able to access protected method clone. Why?**

Some of the Java books imply that all you have to do in order to have your class support clone() is implement the Cloneable interface. Not so. Perhaps that was the intent at some point, but thatï¿½s not the way it works currently. As it stands, you have to implement your own public clone() method, even if it doesnï¿½t do anything special and just calls super.clone().Why is XML such an important development?- It removes two constraints which were holding back Web developments: dependence on a single, inflexible document type (HTML) which was being much abused for tasks it was never designed for; the complexity of full SGML, whose syntax allows many powerful but hard-to-program options. XML allows the flexible development of user-defined document types. It provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web; and it removes the more complex options of SGML, making it easier to program for.

**What is the fastest type of JDBC driver?**

JDBC driver performance will depend on a number of issues:

the quality of the driver code,  
the size of the driver code,  
the database server and its load,  
network topology,  
the number of times your request is translated to a different API.  
In general, all things being equal, you can assume that the more your request and response change hands, the slower it will be. This means that Type 1 and Type 3 drivers will be slower than Type 2 drivers (the database calls are make at least three translations versus two), and Type 4 drivers are the fastest (only one translation).

**How do I find whether a parameter exists in the request object?**

boolean hasFoo = !(request.getParameter(“foo”) == null  
 || request.getParameter

(“foo”).equals(“”));

**or**

boolean hasParameter =  
 request.getParameterMap().contains

(theParameter); //(which works in Servlet 2.3+)  
How can I send user authentication information while makingURLConnection? You will  want to use HttpURLConnection.setRequestProperty and set all the appropriate headers to HTTP authorization.

**What is the difference between TYPE\_SCROLL\_INSENSITIVE , and TYPE\_SCROLL\_SENSITIVE ?**

You will get a scrollable ResultSet object if you specify one of these

ResultSet constants.The difference between the two has to do with whether a result set reflects changes that are made to it while it is open and whether certain methods can be called to detect these changes. Generally speaking, a result set that is TYPE\_SCROLL\_INSENSITIVE does not reflect changes made while it is still open and one that is TYPE\_SCROLL\_SENSITIVE does. All three types of result sets will make changes visible if they are closed and then reopened:

Statement stmt = con.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);  
ResultSet srs =  
stmt.executeQuery(“SELECT NAME, SALARY FROM PERSON”);  
srs.afterLast();  
while (srs.previous())  
{  
String name = srs.getString(“NAME”);  
float salary = srs.getFloat(“SALARY”);  
System.out.println(name + ” ” + salary);  
}

**How do you insert images in Database using JDBC ?**

We can store images in the databse using the BLOB datatype where in the image is stored as a byte stream

**What is Metadata ?**

It is information about one of two things: Database information

(java.sql.DatabaseMetaData), or Information about a specific ResultSet

(java.sql.ResultSetMetaData). Use DatabaseMetaData to find information about your database, such as its capabilities and structure. Use ResultSetMetaData to find information about the results of an SQL query, such as size and types of columns

**What are the new things in JDBC 3.0?**

The new features are:

Savepoint support  
Reuse of prepared statements by connection pools  
Retrieval of auto-generated keys  
Ability to have multiple open ResultSet objects  
Ability to make internal updates to the data in Blob and Clob objects  
Ability to Update columns containing BLOB, CLOB, ARRAY and REF types  
Both java.sql and javax.sql ( JDBC 2.0 Optional Package ) are expected to be included with J2SE 1.4.

**When do you get the message “No Suitable Driver”?**

Often the answer is given that the correct driver is not loaded. This may be the case, but more typically, the JDBC database URL passed is not properly constructed. When a Connection request is issued, the DriverManager asks each loaded driver if it understands the URL sent. If no driver responds that it understands the URL, then the “No Suitable Driver” message is returned.

**Earlier we said we can use same connection but when I try to create multiple Statements on my Connection, only the current Statement appears to be executed. What’s the problem?**

All JDBC objects are required to be threadsafe. Some drivers, unfortunately, implement this requirement by processing tatements serially. This means that additional Statements are not executed until the preceding Statement is completed.

**Explain some tips to improve the performance in a JDBC application?**

These are few points to consider:

Use a connection pool mechanism whenever possible.  
Use prepared statements. These can be beneficial, for example with DB specific escaping, even when used only once.  
Use stored procedures when they can be created in a standard manner. Do watch out for DB specific SP definitions that can cause migration headaches.  
Even though the jdbc promotes portability, true portability comes from NOT depending on any database specific data types, functions and so on.  
Select only required columns rather than using select \* from Tablexyz.  
Always close Statement and ResultSet objects as soon as possible.  
Write modular classes to handle database interaction specifics.  
Work with DatabaseMetaData to get information about database functionality.  
Softcode database specific parameters with, for example, properties files.  
Always catch AND handle database warnings and exceptions. Be sure to check for additional pending exceptions.  
Test your code with debug statements to determine the time it takes to execute your query and so on to help in tuning your code. Also use query plan functionality if available.  
Use proper (And a single standard if possible) formats, especially for dates.  
Use proper data types for specific kind of data. For example, store birthdate as a date type rather than, say, varchar.

**How does the JDBC work?**

The Java Database Connectivity (JDBC) is used to whenever a Java application should communicate with a relational database for which a JDBC driver exists.

Main JDBC classes:

DriverManager. Manages a list of database drivers.  
Driver. The database communications link, handling all communication with the database.  
Connection. Interface with all the methods for contacting a database  
Statement. Encapsulates an SQL statement which is passed to the database to be parsed, compiled, planned and executed.  
ResultSet. The answer/result from a statement. A ResultSet is a list which encapsulates all outgoing results from a given SQL query.