1. **What is JDBC?**

Answer: *JDBC* stands for Java Database Connectivity. It is a Java database connectivity technology from Oracle Corporation. This technology is an *Application Program Interface* (API) for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database.

1. **Describe a general JDBC architecture?**

Answer: *Java Database Connectivity (JDBC) architecture* is an *API* specifying interfaces for accessing relational databases. It helps to connect to a database, send queries and updates to the database, and retrieve and process the results obtained from the database for queries.

1. **What are the common JDBC API components?**

Answer: *The JDBC API* provides the following interfaces and classes −

* DriverManager
* Driver
* Connection
* Statement
* ResultSet
* SQLException.

1. **What is JDBC Driver and DriverManager?**

Answer: A *JDBC driver* is a [software](https://en.wikipedia.org/wiki/Software) component that enables a [Java](https://en.wikipedia.org/wiki/Java_%28programming_language%29) application to interact with a [database](https://en.wikipedia.org/wiki/Database). It translates *API* calls into operations for a specific data source.

*DriverManager* is a software component that loads database drivers and manages the connection between the application and the driver.

1. **Difference between PreparedStatement and Statement.**

Answer:

|  |  |
| --- | --- |
| **Statement** | **PreparedStatement** |
| 1. It is used to implement simple SQL statements with no parameters. | 1. It is used for pre-compiling SQL statements that might contain input parameters. |
| 2. For example: “ResultSet rs = stmt.executequery(“select lastname, firstname from employees”);” | 2. For example: “PreparedStatement ps = conn.prepareStatement(“create table x(id int, name char)”);” |

1. **What is result set object?**

Answer: The results of executing an SQL query are returned in the form of an object that implements the ResultSet interface and contains the table produced by the SQL query. The ResultSet object contains something called a cursor that we can manipulate to refer to any particular row in the resultset.

1. **What is a RowSet?**

Answer: *RowSet* is a wrapper around a *ResultSet* object, an enhanced *ResultSet* object. It maintains a connection to its data source, just as a *ResultSet* object does. As a consequence, a *JDBC* rowset can, for example, be a component in a *Java Swing* application.

1. **What is SQLException?**

Answer: The *SQLException* class and its subtypes provide information about errors and warnings that occur while a data source is being accessed. The base class for exceptions that occur while running JDBC applications is *SQLException*.  
The following information is available from a *SQLException*:

* Text description;
* SQLState;
* Error code;
* A reference to any other exceptions that also occurred.

1. **What is TableModel?**

Answer: The *TableModel* interface specifies the methods which the *JTable* will use to interrogate a tabular data model. The *JTable* can be set up to display any data model which implements the *TableModel* interface with a couple of lines of code:

TableModel myData = new TableModel();

JTable table = new JTable(myData);

1. **What is the difference between executequery() and executeupdate().**

Answer:

|  |  |
| --- | --- |
| **executeQuery** | **executeUpdate** |
| 1. *executeQuery* expects only an SQL statement that generate results. | 1. *executeUpdate* is to execute statements that change the contents of the database rather than returning results. |
| 2. It returns the *ResultSet*. | 2. It returns the number of rows affected by the operations. |

1. **What is DefaultMutableTreeNode & DefaultTreeModel?**

Answer:

|  |  |
| --- | --- |
| **DefaultMutableTreeNode** | **DefaultTreeModel** |
| 1. A *DefaultMutableTreeNode* is a general-purpose node in a tree data structure. | 1. *DefaultTreeModel* is a simple tree data model that uses TreeNodes. |
| 2. For example: “DefaultMutableTreeNode dbNode = new DefaultMutableTreeNode(“authors”);” | 2. For example: “DefaultTreeModel dbTreeModel = new DefaultTreeModel(dbNode);” |

1. **What are the common tasks/steps of JDBC?**

Answer: The common tasks/steps of *JDBC* are:

* Create an instance of a *JDBC* driver or load *JDBC* drivers through *jdbc.drivers*;
* Register a driver;
* Specify a database;
* Open a database connection;
* Submit a query;
* Receive results.