1) What are the JDBC statements?

In JDBC, Statements are used to send SQL commands to the database and receive data from the database. There are various methods provided by JDBC statements such as execute(), executeUpdate(), executeQuery, etc. which helps you to interact with the database.

There is three type of JDBC statements given in the following table.

|  |  |
| --- | --- |
| **Statements** | **Explanation** |
| Statement | Statement is the factory for resultset. It is used for general  purposeaccess to the database. It executes a static SQL query  at runtime. |
| PreparedStatement | The PreparedStatement is used when we need to provide input  parameters to the query at runtime. |
| CallableStatement | CallableStatement is used when we need to access the database  stored procedures. It can also accept runtime parameters. |

### 2) What is the return type of Class.forName() method?

The Class.forName() method returns the object of java.lang.Class object.

3) What are the differences between Statement and PreparedStatement interface?

|  |  |
| --- | --- |
| **Statement** | **PreparedStatement** |
| The Statement interface provides methods to execute queries with the database. The statement interface is a factory of ResultSet; i.e., it provides the factory method to get the object of ResultSet. | The PreparedStatement interface is a  subinterface of Statement. It is used to execute  the parameterized query. |
| In the case of Statement, the query is compiled each time we run the program. | In the case of PreparedStatement, the query is  compiled only once. |
| The Statement is mainly used in the case when we need to run the static query at runtime. | PreparedStatement is used when we need to  Provide input parameters to the query at  runtime. |

### 4) How can we set null value in JDBC PreparedStatement?

By using setNull() method of PreparedStatement interface, we can set the null value to an index. The syntax of the method is given below.

1. **void** setNull(**int** parameterIndex, **int** sqlType) **throws** SQLException

5) What are the benefits of PreparedStatement over Statement?

The benefits of using PreparedStatement over Statement interface is given below.

* The PreparedStatement performs faster as compare to Statement because the Statement needs to be compiled everytime we run the code whereas the PreparedStatement compiled once and then execute only on runtime.
* PreparedStatement can execute Parameterized query whereas Statement can only run static queries.
* The query used in PreparedStatement is appeared to be similar every time. Therefore, the database can reuse the previous access plan whereas, Statement inline the parameters into the String, therefore, the query doesn't appear to be same everytime which prevents cache reusage.

### 6) What are the differences between execute, executeQuery, and executeUpdate?

|  |  |  |
| --- | --- | --- |
| **execute** | **executeQuery** | **executeUpdate** |
| The execute method can be used for any SQL statements(Select and Update both). | The executeQuery method can be used only with the select statement. | The executeUpdate method can be  used to update/delete/insert  operations in the database. |
| The execute method returns a boolean type value where true indicates that the ResultSet s returned which can later be extracted and false indicates that the integer or void value is returned. | The executeQuery() method returns a ResultSet object which contains the data retrieved by the select statement. | The executeUpdate() method  an integer value representing the  number of records affected where  0 indicates that query returns  nothing |

### 7) What are the different types of ResultSet?

ResultSet is categorized by the direction of the reading head and sensitivity or insensitivity of the result provided by it. There are three general types of ResultSet.

|  |  |
| --- | --- |
| Type | Description |
| ResultSet.TYPE\_Forward\_ONLY | The cursor can move in the forward direction  only. |
| ResultSet.TYPE\_SCROLL\_INSENSITIVE | The cursor can move in both the direction  (forward and backward). The ResultSet is not  sensitive to the changes made by the others  to the database. |
| ResultSet.TYPE\_SCROLL\_SENSITIVE | The cursor can move in both the direction.  The ResultSet is sensitive to the changes made  by the others to the database. |

### 8) What are the differences between ResultSet and RowSet?

|  |  |
| --- | --- |
| ResultSet | RowSet |
| ResultSet cannot be serialized as it maintains the connection with the database. | RowSet is disconnected from the database and can  be serialized. |
| ResultSet object is not a JavaBean object | ResultSet Object is a JavaBean object. |
| ResultSet is returned by the executeQuery() method of Statement Interface. | Rowset Interface extends ResultSet Interface and  returned by calling  the RowSetProvider.newFactory().createJdbcRowSet()  method. |
| ResultSet object is non-scrollable and non-updatable by default. | RowSet object is scrollable and updatable by default. |

### 9) What is the role of the JDBC DriverManager class?

The DriverManager class acts as an interface between user and drivers. It keeps track of the drivers that are available and handles establishing a connection between a database and the appropriate driver. The DriverManager class maintains a list of Driver classes that have registered themselves by calling the method DriverManager.registerDriver().

### 10) What are the functions of the JDBC Connection interface?

The **Connection interface** maintains a session with the database. It can be used for transaction management. It provides factory methods that return the instance of Statement, PreparedStatement, CallableStatement, and DatabaseMetaData.

### 11) What does the JDBC ResultSet interface?

The ResultSet object represents a row of a table. It can be used to change the cursor pointer and get the information from the database. By default, ResultSet object can move in the forward direction only and is not updatable. However, we can make this object to move the forward and backward direction by passing either TYPE\_SCROLL\_INSENSITIVE or TYPE\_SCROLL\_SENSITIVE in createStatement(int, int) method.

### 12) What does the JDBC ResultSetMetaData interface?

The ResultSetMetaData interface returns the information of table such as the total number of columns, column name, column type, etc.

### 13) What does the JDBC DatabaseMetaData interface?

The DatabaseMetaData interface returns the information of the database such as username, driver name, driver version, number of tables, number of views, etc.

### 14) Which interface is responsible for transaction management in JDBC?

The **Connection interface** provides methods for transaction management such as commit(), rollback() etc.

15) What is batch processing and how to perform batch processing in JDBC?

By using the batch processing technique in JDBC, we can execute multiple queries. It makes the performance fast. The java.sql.Statement and java.sql.PreparedStatement interfaces provide methods for batch processing. The batch processing in JDBC requires the following steps.

* Load the driver class
* Create Connection
* Create Statement
* Add query in the batch
* Execute the Batch
* Close Connection

16) What is the JDBC Rowset?

JDBC Rowset is the wrapper of ResultSet. It holds tabular data like ResultSet, but it is easy and flexible to use. The implementation classes of RowSet interface are as follows:

* JdbcRowSet
* CachedRowSet
* WebRowSet
* JoinRowSet
* FilteredRowSet

### 17) What is the major difference between java.util.Date and java.sql.Date data type?

The major difference between java.util.Date and java.sql.Date is that, java.sql.Date represents date without time information whereas, java.util.Date represents both date and time information.

### 18) What does JDBC setMaxRows method do?

The setMaxRows(int i) method limits the number of rows the database can return by using the query. This can also be done within the query as we can use the limit cause in MySQL.

19) What is stored procedure? What are the parameter types in stored procedure?

* Stored procedure is a group of SQL queries that are executed as a single logical unit to perform a specific task. Name of the procedure should be unique since each procedure is represented by its name.

Stored procedures are called using CallableStatement class available in JDBC API

Three types of parameters are provided in the stored procedures. They are:

* **IN**: It is used for passing the input values to the procedure. With the help of setXXX() methods, you can bind values to IN parameters.
* **OUT**: It is used for getting the value from the procedure. With the help of getXXX() methods, you can obtain values from OUT parameters.
* **IN/OUT**: It is used for passing the input values and obtaining the value to/from the procedure. You bind variable values with the setXXX() methods and obtain values with the getXXX() methods

20) What do you mean by DatabaseMetaData and why we are using it?

* DatabaseMetaData is an interface that provides methods to obtain information about the database.
* We can use this for getting database-related informations, such as database name, database version, driver name, the total number of tables or views, etc.

21) What are the differences between ODBC and JDBC?

| **ODBC(Open Database Connectivity)** | **JDBC(Java Database Connectivity)** |
| --- | --- |
| ODBC can be used for languages like C, C++, Java, etc. | JDBC is used only for the Java language |
| We can use ODBC only for the Windows platform, thus it is platform-dependent. | We can use JDBC on any platform, thus it is platform-independent |
| Most of the ODBC Drivers developed in native languages like C, C++ | JDBC drivers are developed using the Java language |
| It is not recommended to use ODBC for Java applications, because of low performance due to internal conversion. | It is highly recommended to use JDBC for Java applications because there are no performance issues. |
| ODBC is procedural. | JDBC is Object Oriented. |

22). What is Rowset?

* A RowSet is an object that encapsulates a row set from either JDBC result sets or tabular data sources such as files or spreadsheets. It supports component-based development models like JavaBeans, with the help of a standard set of properties and event notifications.
* The advantages of using RowSet are:
  + It is easier and flexible to use.
  + It is Scrollable and Updatable by default.

### 23) What is the difference between Statement and PreparedStatement?

| **Statement** | **PreparedStatement** |
| --- | --- |
| The query is compiled every time we run the program. | The query is compiled only once. |
| It is used in the situation where we need to run the SQL query without providing parameters at runtime. | It is used when we want to give input parameters to the query at runtime. |
| Performance is less compared to PreparedStatement. | Provides better performance than Statement, as it executes the pre-compiled SQL statements. |
| It is suitable for executing DDL statements such as CREATE, ALTER, DROP and TRUNCATE. | It is suitable for executing DML statements such as INSERT, UPDATE, and DELETE. |
| It cannot be used for storing/retrieving images and files in the database. | It can be used for storing/retrieving images and files in the database. |
| It executes static SQL statements. | It executes pre-compiled SQL statements. |
| Less secured as it enforces SQL injection. | More secured as they use bind variables, which can prevent SQL injection. |

### 24) Explain the difference between execute(), executeQuery() and executeUpdate() methods in JDBC.

| **execute()** | **executeQuery()** | **executeUpdate()** |
| --- | --- | --- |
| It can be used for any SQL statements. | It is used to execute SQL Select queries. | It is used to execute the SQL statements such as Insert/Update/Delete which will update or modify the database data. |
| It returns the boolean value TRUE if the result is a ResultSet object and FALSE when there is no ResultSet object. | It returns the ResultSet object which contains the data retrieved by the SELECT statement. | It returns an integer value which represents the number of affected rows where 0 indicates that the query returns null. |
| Used for executing both SELECT and non-SELECT queries. | Used for executing only the SELECT Query. | Used for executing only a non-SELECT query. |

The execute() method is used in the situations when you are not sure about the type of statement else you can use executeQuery() or executeUpdate() method.

### 25). How to use JDBC API to call Stored procedures?

Stored procedures are a set of SQL queries that are compiled in the database and will be executed from JDBC API. For executing Stored procedures in the database, JDBC CallableStatement can be used. The syntax for initializing a CallableStatement is:

CallableStatement cs = con.prepareCall("{call insertEmployee(?,?,?,?,?)}");

stmt.setInt(1, id);

stmt.setString(2, name);

stmt.setString(3, role);

stmt.setString(4, address);

stmt.setString(5, salary);

//registering the OUT parameter before calling the stored procedure

cs.registerOutParameter(5, java.sql.Types.VARCHAR);

cs.executeUpdate();

We must register the OUT parameters before executing the CallableStatement.

26) Explain the benefits of PreparedStatement over Statement.

Benefits of PreparedStatement over Statement interface are:

* It performs faster compared to the Statement because the Statement needs to be compiled each time when we run the code whereas the PreparedStatement is compiled once and then executed only on runtime.
* It can execute parametrized queries. But Statement can only run static queries.
* The query used in PreparedStatement looks similar each time, so the database can reuse the previous access plan. Statement inline the parameters into the string, so the query doesn’t look to be the same every time which prevents reusage of cache.