JAVA DATABASE CONNECTIVITY (JDBC)

JDBC is a Java API used to make things easy. Which is used to execute common SQL queries and operate on the database.

JDBC API is Java API that can access any kind of tabular data that is stored in relational database. It uses JDBC Drivers to connect with database.

The database is used to store the information in such a way that it can be retrieve from the database whenever needed. It stores data in tables in a row and column manner. The tables are relations; it collects objects of the same type.

JDBC helps to write Java applications that manages activities:

1) *To connect to database.*
2) *Send queries and update statements to the database.*
3) *Retrieve and process the data received from the database.*

DriverManager class is used to establish the connection with database using the getConnection() method. It loads JDBC Driver in java application to establish the connection. Connection is an interface.

```
Connection conn = DriverManager.getConnection();
```

The connection object is needed to create Statement object. The Statement is an interface.

```
Statement stm = conn.createStatement();
```

Statement object is used to create ResultSet object using executeQuery() method.

```
ResultSet rs = stm.executeQuery();
```

Components of JDBC:

JDBC API: The JDBC API provides methods and interfaces to access relational data using java programming language. Using API also allowed to execute SQL queries, retrieve data, and process the data.

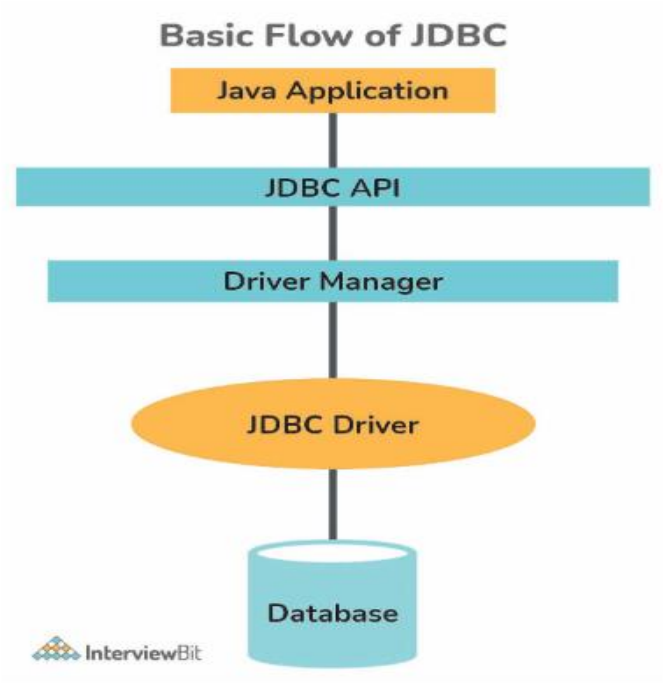
JDBC DriverManager: The JDBC DriverManager class defines objects which can connect to JDBC driver. It loads JDBC Driver in java applications for establishing connection with the database.

JDBC Test Suite: It is used to test the operations like updating, deleting, inserting etc., being performed by the JDBC Drivers.

JDBC-ODBC Bridge Drivers: It will connect database drivers to the database. JDBC-ODBC bridge interprets JDBC method call to the ODBC function call.

JDBC ARCHITECTURE:

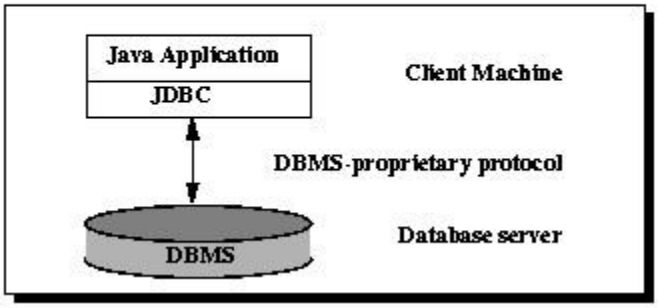
JDBC API supports both *two tier* and *three tier* processing model for database access.



1) Two tier Architecture:

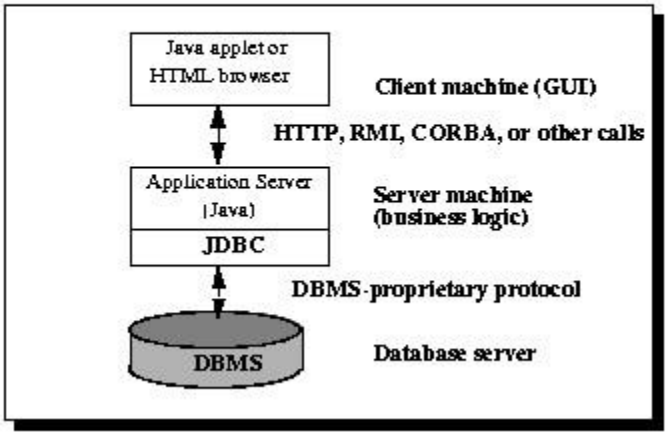
- In Two tier model the java applications directly communicate with data source (database).
- JDBC Drivers helps applications to communicate with the data source.
- A user's commands are delivered to the data source (database) and results are sent back to the user.

Figure 1: Two-tier Architecture for Data Access.



2) Three tier Architecture:

Figure 2: Three-tier Architecture for Data Access.



- In three tier architecture model the ***middle tier*** exists in between the user and the database.
- The middle tier acts as a service provider which receives the commands from the user and sends those to the database and vice versa.
- Three tier model is more effective than two tier because it has a middle tier.
- Advantage is it maintains control over the data access.

STEPS TO PROCESS SQL STATEMENT WITH JDBC:

1) *Establish connection.*
2) *Create a statement.*
3) *Execute the query.*
4) *Process the ResultSet object.*

5) *Close the connection.*

1. Establishing connection:

First, need to establish connection with the data source. Data source may be any database, or any other data source which needs to be use or any external file that corresponds to Driver.

- DriverManager: It is an implementation class; it connects an application with the data source (database) which is specified by the database URL.
- Connection is done using method DriverManager.getConnection().

2. Create statement:

Statement is an interface that represents SQL statement. Statement will generate ResultSet objects, which is a table of data. To create a Statement object, Connection object is needed using createStatement().

```
Coonection conn = DriverManager.getConnection(url, username, password);

Statement stm = conn.createStatement();
```

There are 3 different kinds of statements.

1. ***Statement****:* Statement used to implement simple SQL statements with no parameters.
2. ***PreparedStatement****:* Extends Statement, used for precompiling SQL statements that might contain input parameters.
3. ***CallableStatement****:* Extends PreparedStatement, used to execute stored procedures that may contain both input output parameters.

3. Execute Queries:

Queries can be executed by calling execute() method of the Statement interface such as the following

```
ResultSet rs = stm.execute(query);
```

- ***execute()*** : Returns true if the first object that the query returns ResultsSet object. This method is useful when query could return one or more ResultSet objects. Return type is Boolean.
- ***executeQuery()*** : Returns one ResultsSet object. Return type is ResultSet.

- ***executeUpdate()*** : Returns an integer representing the number of rows affected by the SQL statement. This method is useful when dealing with INSERT, DELETE OR UPDATE SQL statements. Return type is int.

4. Processing the ResultSet Object:

- ResultSet is an interface that provides methods for retrieving and manipulating results of executed query.
- We can access the data in the ResultSet through cursor.
- The cursor is not a database cursor.
- The cursor is a pointer that points to one row of the data in the ResultSet object.
- Initially the cursor is positioned before the first row.
- The next() returns false when cursor is positioned after the last row.
- The next() is used to move forward to the rows.

```
ResultSet rs = stm.executeQuery(query);

While(rs.next()){

        // doSomething

}
```

5. Close the Connection:

After completed using the Connection, Statement, and ResultSet object close the connection by calling the close method. It is necessary to release the data source connection after the usage.

- The resources must be closed regardless of whether the SQLException is handled or not.
- It is important to avoid resource leakage.
- Resource leak is a type of bug that occurs when resource is not properly released. This leads to several problems like system performance, increased memory usage, and security vulnerability.

QUESTIONS:

1. What is JDBC Driver?

The JDBC Driver is a software component(API) having various classes and interfaces, that enables java application to interact with database. It acts as a mediator in between java application and database.

2. Explain PreparedStatement?

- PreparedStatement is more convenient to send SQL statements to the database.
- When executing the statement many times the PreparedSatement is used to reduce the time for executing the statement.
- The advantage of using PreparedStatement is that in most cases SQL statement is sent to DBMS, where it is compiled.
- The PreparedStatement object contains not just SQL statement, but a SQL statement that has been precompiled.
- This means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement SQL statement without having to compile it first.
- PreparedStatement is used when we need to give input data to the query at run time and also if we want to execute query many times.

3. Explain Stored Procedures? What are parameter types in stored procedure?

Stored Procedure is a group of SQL statements that form single unit to perform operations, and they are encapsulate a set of operations or queries to execute on a database.

Stored Procedures can be compiled and executed with different parameters and results, and they can have any combination of input, output or both input/output parameters. Stored procedures are call using the CallableStatement.

In stored procedure there are three types of parameters are present.

- IN: It is used for passing the input values to procedure. With the help of setXXX() method.
- OUT: Is used for getting the values form the procedure. With the help of getXXX() method.
- IN/OUT: It is used for passing input values, and obtaining the values from the procedure using the setXXX() and getXXX() methods.

4. What is JDBC Driver?

The JDBC Driver is a software component(API) contains various classes and interfaces to connect with database. To connect to individual database, JDBC requires particular drivers for each specific database.

5. Which datatype is used for storing images and files in database table?

CLOB: is used to store files in the database. It stores the character type of data.

BLOB: is used to store binary type of data. Using this type can store images, videos and audio type of data.

6. What is DatabaseMetaData?

- DatabaseMetaData is an interface that provides methods to obtain information about the database.
- Information like database name, database version, driver name, total number of tables or views, et.

7. List out the differences between JDBC and ODBC?

ODBC(OPEN DATABASE CONNECTIVITY)	***JDBC(JAVA DATABASE CONNECTIVITY)***
Used for programing languages like C, C++, Java, etc.	Used only for Java language.
ODBC is platform dependent.	JDBC is platform independent.
Most of the ODBC drivers are developed in native languages like C, C++	JDBC drivers are developed using Java language.
Not recommended to use ODBC for java application because of low performance.	JDBC is recommended for Java application.
ODBC is procedural.	JDBC is object oriented.

8. What is RowSet?

- RowSet is an object that encapsulates a row set from either JDBC result sets or tabular data source.
- It supports component based development models like JavaBeans with the help of standard set of properties and event notification.
- The advantage of using the RowSet are is easier, flexible, scrollable and updatable by default.

9. What is Serialization and Deserialization?

Serialization:

- Serialization is the process of the converting a ResultSet object into a stream of bytes that can be stored in file, database, or transmitted over the network.
- To serialize the ResultSet object using the ObjectOutputStream class.
- By using the method called writeObject() which takes argument and writes it to stream of bytes.

Deserialization:

- It is a process of converting a byte stream back into an original object is called deserialization.
- Deserialization is used to retrieve the object that have been stored in the database.
- To desterilize the object using the getObject() method.

```
Object obj = resultSetObj.getObject("serializedObject");
```

10. Difference between RowSet and ResultSet?

ResultSet	***RowSet***
ResultSet can't be serialized because it handles connection to database.	RowSet is disconnected from the database so is can be serialized.
By default ResultSet object is non scrollable and non-updatable.	RowSet is scrollable and updatable.
ResultSet object is a JavaBean object.	RowSet object is a JavaBean object.
ResultSet is returned by executeQuery() of the Statement interface.	RowSet extends the ResultSet interface and is returned by calling the RowSetProvider.createFactory() .createJdbcRowSet()
It is difficult to pass ResultSet from one class to another has it has connected to database.	It is easier to pass RowSet from one class to another has it has not connected to database.

11. Explain the types of ResultSet?

There are three types of ResultSet are there to control the cursor pointing in forward, backward, and in a particular row. If do not declare any type, then TYPE_FORWARD_ONLY will call.

1) *ResultSet.TYPE_FORWARD_ONLY:* Using this the cursor can only move forward from start to end in the ResultSet.
2) *ResultSet.TYPE_SCROLL_INSENSITIVE:* Using this cursor can move both forward and backward direction. Here, the result set is insensitive to the changes done in the database by others, that occurs after the result set was created.
3) *ResultSet.TYPE_SCROLL_SENSITIVE:* Using this cursor can move on both the directions, and result set is sensitive to changes made to the database by others, that occurs after the result set was created.

12. Difference between Statement and PreparedStatement.

Statement	***PreparedStatement***
Query is compiled every time we run the program.	Query is compiled only once.
It is used when executing query without providing the parameters at runtime.	It is used when query need input parameters.
Performance is less.	Performance is more.
It is suitable for executing DDL statements such as CREATE, ALTER, DROP and TRUNCATE.	It is suitable for DML statements such as INSERT, UPDATE, and DELETE.
It executes static SQL statements.	It executes pre-compiled SQL statements.