**JDBC:-**

JDBC stands for Java database connectivity.

DataBase

JDBC req

Servlet Req

Front End

Middleware

Frf

**API:-**

Application programming interface. It is group of classes and interfaces which work together to provide additional feature to a programming language.

**JDBC:-**

JDBC stands for Java database connectivity.

JDBC is a JAVA API which works in Backend to establish connection with to database and to convert java object to database understandable value and Database values to java understandable value.

**There are 7 steps in JDBC.**

**7 steps of JDBC:-**

1. **Importing the packages**
2. **Load and register Driver**
3. **Establish the Connection**
4. **Create the statement**
5. **Execute to the statement**
6. **Process the result**
7. **Close the connection**

**Importing the Packages:-**

1. To import the required the files or packages required to run a JDBC program, we need to download jar file of database connector.
2. This Downloaded jar file should be added to java build path.

**Driver:-**

1. It is a third party software (DB software) Which is used to teach the type of database we are trying to access using java program.

In this example we are using type four driver i.e Database dependent driver.

1. To load and register the driver program with java class we will use **forName()** method form class **Class.**

**forName():-**

It is static method **from class Class** which is used to instantiate and retrieve the details of the class whose fully qualified name is passed as a string argument to the method.

**Example:-**

**Class.*forName*("com.mysql.cj.jdbc.Driver");**

**Establish the connection:-**

In this step the java program connected with database.

To get a connection method from **Drivermanger** class and provide String url,

String username, string password) as an argument **to getconnection().**

**Example:-**

Connection con= DriverManager.*getConnection*("jdbc:mysql://localhost:3306/m2e2jdbcday1","root","tiger");

Connection-: Interface

DriverManager -class

**1.getconnection(String url,String user,String password):-**

It is static method from driver manger class it is a overloaded method where first get connection method.

1.getconnection(String url, String username, String Password).

**Example:**

Connection con= DriverManager.*getConnection* ("jdbc: mysql : //localhost:3306/firstdb","root","tiger");

**2.get connection(url):-**

method accept only string url and we need provide username and password as query String.

**Example:**

Driver driver=new Driver();

DriverManager.registerDriver(driver);

DriverManager.getConnection("jdbc:mysql://localhost:3306/firstdb?user=root&password=root");

**3.getconnection(String url, properties info):-**

getconnection() accept String url and properties info where username, password of the database is provided as an information in properties file. All of these get connection method returns object reference of implementing class of **Connection Interface.**

**Example:**

Driver driver = new Driver();

DriverManager.registerDriver(driver);

FileInputStream f=new FileInputStream("myDbInfo.properties");

Properties properties=new Properties();

properties.load(f);

String url="jdbc:mysql://localhost:3306/firstdb";

Connection con=DriverManager.getConnection(url,properties);

DriverManager.*registerDriver*(driver);

FileInputStream f=**new** FileInputStream("myDbInfo.properties");

Properties properties=**new** Properties();

properties.load(f);

String url="jdbc:mysql://localhost:3306/firstdb";

Connection con=DriverManager.*getConnection*(url,properties);

Sjsjs

User:root

Password:tiger

myDbInfo.properties

**Statement:-**

* It is an interface from java.sql.package. Non-static methods from connection interface called create statement returns the object reference of implementing class of statement.
* Statement as function which are use to interact with database.

Example:-

Execute()

Statement statement=connnection.createStatement();

String query="INSERT INTO Student(sid,sname,email,age) VALUES (101,'Mirza','abc@gmail.com',22)";

boolean res=statement.execute(query);

**Excute method():-**

* It is non-static method of statement interface which accept database query as String argument.
* This execute() can be used for all four **CRUD** operation and for select query it will return true and for others it will return false.

**ExecuteUpdate():-**

* It is non-static method from statement interface which accept database query as a string argument.
* This method can be only used for insert, update and delete query.
* The return type of this method is Integer where this int value shows number of rows that got updated.

**ExecuteQuery():-**

**ResultSet:-**