**JDBC**-

* With Core Java knowledge we can develop Stand Alone Applications.
* The Applications which are running on a Single Machine are called *Stand Alone Applications.*

Eg: Calculator, MS Word, Any Core Java Application

* If we want to develop Web Applications then we should go for Advanced Java.
* The Applications which are providing Services over the Web are called *Web Applications.*

Eg: gmail.com, facebook.com, irctc.com

* In Java we can develop Web Applications by using the following Technologies...

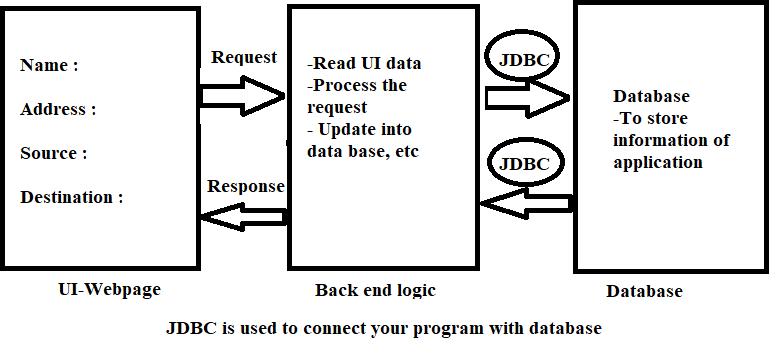
JDBC

Servlets JSP's, etc…

**Why?**

Suppose I want to connect java program to the database then how can I do this?

By using JDBC, we can achieve this.



## Components of JDBC :

1. **Driver (Translator):**

To convert Java specific calls into Database specific calls and Database specific calls into Java calls.

## Connection (Road):

By using Connection, Java Application can communicate with Database.

## Statement (Vehicle):

By using Statement Object, we can send our SQL Query to the Database and we can get Results from Database.

## ResultSet:

ResultSet holds Results of SQL Query.

How to add the MySQL jar file into project.

Right click on Project->Build Path->Configure Build Path->Click on Libraries -> Add External Jar->Select Jar File-> Click on Apply and Close or Apply button.

JDBC stands for Java Database Connectivity.

It is an API for the Java programming language.

It allows the client to access the database and also establishes how the client may do so. It can work on a number of operating systems or platforms, like Windows, Mac, etc. It is basically a connection between the database and the application.

To connect Java application with the MySQL database, we need to follow 5 following steps.

1. **Driver class:**The driver class for the MySQL database is **com.mysql.jdbc.Driver**.
2. **Connection URL:**The connection URL for the mysql database is **jdbc:mysql://localhost:3306/test**

jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and test is the database name.

1. **Username:**The default username for the mysql database is **root**.
2. **Password:**It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.
3. **Create a table -**in the mysql database, but before creating table, we need to create database first.

**Steps-**

Step 1: Load the Driver class

Step 2: Establish the connection

Step 3: Create the statement

Step 4: Prepare the SQL statement

Step 5: Submit the SQL statement to Database

Step 6: Process the Results

Step 7: Release the Resources

MYSQL query for table creation-

CREATE TABLE `user` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`LastName` varchar(255) DEFAULT NULL,

`FirstName` varchar(255) DEFAULT NULL,

`Address` varchar(255) DEFAULT NULL,

`City` varchar(255) DEFAULT NULL,

`Salary` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

);

**Example-1** Program for insert the student data using statement through JDBC.

**package** com.operation;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

//Insert the student data using statement

**public** **class** InsertData {

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException, SQLException {

**try** {

String sql = "insert into user(lastName,firstName,Address,City,Salary)"

+ "values('pawar','ram','sangavi','pune',5000)";

Class.*forName*("com.mysql.jdbc.Driver"); // load the driver

// to establish the connection

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

// create the sql statement

Statement statement = con.createStatement();

// submit the sql statement to database..

//statement.executeUpdate(sql);

statement.execute(sql);

System.***out***.println("Insertion successfully...");

// close the resources.

con.close();

statement.close();

} **catch** (Exception e) {

System.***out***.println(e);

}

}

}

**Output-**

Insertion successfully message displayed on screen and data will be stored into database.

MYSQL query for table creation-

CREATE TABLE `employee` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(255) NOT NULL,

`password` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

)

**Example-2** Program for insert the student data using prepared statement.

**package** com.operation;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

//insert the data using preparedstatement

**public** **class** InsertStudent {

**public** **static** **void** main(String[] args) {

**try** {

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

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

PreparedStatement stmt = con.prepareStatement("insert into employee(username,password)values(?,?)");

stmt.setString(1, "admin"); //1 first parameter in query.

stmt.setString(2, "guest");

**int** i=stmt.executeUpdate();

System.***out***.println("Record is inserted."+i);

con.close();

stmt.close();

} **catch** (Exception e) {

e.getMessage();

}

}

}