**How many editions of Java are there?**

* **Java standard edition—standalone applications(core java/jdbc)**
* **Java enterprise edition(jsp/servlets)**
* **Java micro edition(mobile and embedded applications)**

**What is JDBC ?  
Is is the one and only API in the world of java to connect to the database.**

**API:**

**Collection of packages with a dedicated functionality or basically any kind of specification.**

**They help us in communicating between two applications.**

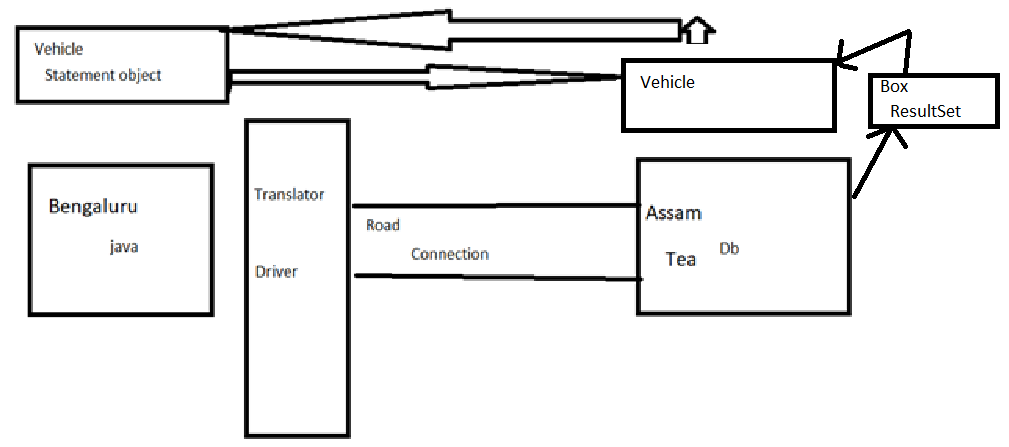
**J2EE has 3 API:**

**JDBC—4.2**

**SERVLETS—**

**JSP—**

**Let me give you an example and with that example you would be able to understand what actually is JDBC and what is the use also how it is used.**

****

**So here, to convert java statements to DB statements we need a translator,   
1) that is nothing but the Driver.**

**2) To reach the database some network or connection is required that is nothing but your Connection.**

**Now, we have to send one SQL query to the database and database engine has to execute that sql query and send us the result right?   
According to the diagram that is a vehicle. So, in JDBC terms this particular vehicle is known as**

**3) Statement object: purpose is to send our SQL queries to the database and to bring back the results from DB to our Java program and the our Java app can now access the statement object.**

**Now again, once our vehicle reaches the DB, db engine will execute the query and produce a result right. After producing the result the DB will do what put the result inside a box**

**4) which is known as the Result. And then the statement object which is our vehicle will bring the result in it and our java program can access the result by using the statement object.**

**FIVE STEPS TO USE JDBC:**

**1. Load the driver--> two ways: Create the object of driver and invoke the registerDriver(passing the reference ) method persent in DriverManager class.**

**: Use the Class.forName(Driver class fully qualified name )**

**2. Get the database connection via driver. Connection connection= DriverManager.getConnection(dburl);**

**3. Issue the Sql queries via connection.**

**4. process the result returned by sql.**

**5. close all JDBC resources.**

**As discussed earlier JDBC is an API/Specification. Who provides that specifications? Java vendors, sun microsystems now oracle.  
And now who is responsible to implement these specifications ? DataBase vendors. So, these database vendors provides there own Drivers. These are the drivers we have to load. So, if we are using oracle DB we will have a diff driver, for mysql a diff driver and so on.**

**JDBC Features:**

1. **JDBC is a standard API, and it is DB independent.**

**DRIVERS in JDBC:**

**>External component.**

**>Should be provided by the DB vendors.**

**>It is DB dependent.**

**>Having drivers is mandatory.**

**JAR Files:**

**Collection of .class files, packages and other resources.**

**URL:**

**WebUrl**

**smttUrl**

**tcpUrl**

**udpUrl**

**dbUrl: <protocol>:<subprotocol>:<subname>**

**|**

**JDBC(case insensitive)**

**DBURL: jdbc:mysql://localhost:3306/dbname? user=root & password= root;**

**STATIC QUeries: may or maynot have conditions, if the query has conditions the conditions should be hardcoded.**

**We need a Statement object. We can get it by Connection.createStatement();**

**Statement statement= connection.createStatement();**

**DYNAMIC queries: must have condition and one or more than one value should be inserted at the runtime using placeholder(?)**

**We need a preparedstatement object, can get it by Connection.prepareStatement();**

**PreparedStatement prepare= connection.prepareStatement();**

**for select queries we need executeQuery(), return type ResultSet**

**ResultSet resultSet = statement.executeQuery();**

**for non select query we need to use executeUpdate and return type int.**

**int result = prepare.executeUpdate();**

**PROPERTIES FILES:**

**Properties class extends HashTable and present in java.util.**

**Static Insert**

**package** com.te.jdbcapp;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** StaticInsert {

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

Connection connection = **null**;

Statement statement = **null**;

**try** {

//Step 1

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

// Step 2

String dbUrl = "jdbc:mysql://localhost:3306/ems";

connection = DriverManager.*getConnection*(dbUrl, "root", "root");

// Step 3

String query = "Insert into primary\_info values" + "(30, 'Nilim', '1992-10-02', '2869273973')";

statement = connection.createStatement();

**int** result = statement.executeUpdate(query);

// Step 4

**if** (result != 0) {

System.***out***.println(result + " No of rows affected");

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

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (connection != **null**) {

**try** {

connection.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**if** (statement != **null**) {

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

}

Static read:

**package** com.te.jdbcapp;

**import** java.sql.\*;

**import** java.util.ArrayList;

**public** **class** StaticRead {

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

Connection connection = **null**;

Statement statement = **null**;

ResultSet resultSet = **null**;

**try** {

// Load the driver

// Driver driver= new com.mysql.jdbc.Driver();

// DriverManager.registerDriver(driver);

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

//step 1 end

//Get the connection via Driver

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

connection = DriverManager.*getConnection*(dbUrl);

//step 2 end

//Issue sql queries via connection

String query = "select \* from primary\_info";

statement = connection.createStatement();

resultSet = statement.executeQuery(query);

// end of step 3

//'Process the result returned by sql

**while** (resultSet.next()) {

System.***out***.println("Id = " + resultSet.getInt(1));

System.***out***.println("name = " + resultSet.getString(2));

System.***out***.println("dob = " + resultSet.getDate(3));

System.***out***.println("Phone no = " + resultSet.getString(4));

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**try** {

**if** (connection != **null**) {

connection.close();

}

**if** (statement != **null**) {

statement.close();

}

**if** (resultSet != **null**) {

resultSet.close();

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

}

To read data by storing it in a object of employeeDetails class:

**package** com.te.jdbcapp;

**import** java.util.Date;

**public** **class** EmployeeDetails {

**private** **int** id;

**private** String name;

**private** Date dob;

**private** String phno;

@Override

**public** String toString() {

**return** "EmployeeDetails [id=" + id + ", name=" + name + ", dob=" + dob + ", phno=" + phno + "]";

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Date getDob() {

**return** dob;

}

**public** **void** setDob(Date dob) {

**this**.dob = dob;

}

**public** String getPhno() {

**return** phno;

}

**public** **void** setPhno(String phno) {

**this**.phno = phno;

}

}

**package** com.te.jdbcapp;

**import** java.sql.\*;

**import** java.util.ArrayList;

**public** **class** EmployeeDetailsRead {

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

ArrayList<EmployeeDetails> employeeDetails = **new** ArrayList<EmployeeDetails>();

Connection connection = **null**;

Statement statement = **null**;

ResultSet resultSet = **null**;

**try** {

// Load the driver

// Driver driver= new com.mysql.jdbc.Driver();

// DriverManager.registerDriver(driver);

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

//step 1 end

//Get the connection via Driver

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

connection = DriverManager.*getConnection*(dbUrl);

//step 2 end

//Issue sql queries via connection

String query = "select \* from primary\_info";

statement = connection.createStatement();

resultSet = statement.executeQuery(query);

// end of step 3

//'Process the result returned by sql

**while** (resultSet.next()) {

EmployeeDetails emplDetails = **new** EmployeeDetails();

emplDetails.setId(resultSet.getInt(1));

emplDetails.setName(resultSet.getString(2));

emplDetails.setDob(resultSet.getDate(3));

emplDetails.setPhno(resultSet.getString(4));

employeeDetails.add(emplDetails);

}

**for** (EmployeeDetails employeeDetails2 : employeeDetails) {

System.***out***.println(employeeDetails2);

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (connection != **null**) {

**try** {

connection.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**if** (resultSet != **null**) {

**try** {

resultSet.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**if** (statement != **null**) {

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

}

Update operation :

**package** com.te.jdbcapp;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** StaticUpdate {

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

Connection connection = **null**;

Statement statement = **null**;

// Step 1

**try** {

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

// Step 2

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

connection = DriverManager.*getConnection*(dbUrl);

// Step 3

String query = "update primary\_info set phno= '998987654' where id=20";

statement = connection.createStatement();

**int** result= statement.executeUpdate(query);

// step 4

**if**(result!=0) {

System.***out***.println(result + " No of rows affected");

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

}

} **catch** (Exception e) {

e.printStackTrace();

}

// step 5

**finally** {

**if**(connection!=**null**) {

**try** {

connection.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(statement!=**null**) {

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

}

}

Static Delete:

**package** com.te.jdbcapp;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** StaticDelete {

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

Connection connection = **null**;

Statement statement = **null**;

//Step 1

**try** {

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

//Step 2

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

connection = DriverManager.*getConnection*(dbUrl);

//Step 3

String query = "delete from primary\_info where id=20";

statement = connection.createStatement();

**int** result = statement.executeUpdate(query);

//Step 4

**if**(result!=0) {

System.***out***.println(result + "rows affected");

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

}

} **catch** (Exception e) {

e.printStackTrace();

}

**finally** {

**if**(connection!=**null**) {

**try** {

connection.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

**if**(statement!=**null**) {

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

}

}

How to store data in a properties file:

package com.te.jdbcapp;

import java.io.FileOutputStream;

import java.util.Properties;

public class PropertiesExample {

public static void main(String[] args) {

try {

//Create a property file with properties extension

//To create a file we need to use FileOutputStream.

FileOutputStream fileOutputStream = new FileOutputStream("directory.properties");

//Create a Properties class object

Properties properties = new Properties();

//Use the reference of the properties class and call the setProperty(<key>, <values>)

properties.setProperty("Nilim", "9800495959");

properties.setProperty("Naveen", "8849093933");

//Use the reference of properties and call method store(object of FileOutputStream, comments)

properties.store(fileOutputStream, "My first property file");

System.out.println("Success");

} catch (Exception e) {

e.printStackTrace();

}

}

}

How to read from the properties file?

**package** com.te.jdbcapp;

**import** java.io.FileInputStream;

**import** java.util.Properties;

**public** **class** ReadPropertiesFileExample {

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

**try** {

FileInputStream fileInputStream=**new** FileInputStream("directory.properties");

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

properties.load(fileInputStream);

System.***out***.println(properties.getProperty("Nilim"));

System.***out***.println(properties.getProperty("Naveen"));

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

----------------Start from here today----------------

Tell about PreparedStatement and prepareStatement()

DYNAMIC QUERIES:

Dynamic insert using scanner:

**package** com.te.jdbcfirst.dynamic;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.util.Scanner;

**public** **class** DynamicInsert {

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

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter the Id : ");

**int** id = scanner.nextInt();

System.***out***.println("Enter the name : ");

String name = scanner.next();

System.***out***.println("Enter the date : ");

String date = scanner.next();

System.***out***.println("Enter the address : ");

String address = scanner.next();

Connection connection= **null**;

PreparedStatement preparedStatement = **null**;

**try** {

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

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

connection = DriverManager.*getConnection*(dbUrl);

String query = "Insert into emp\_info values( ?, ? , ? , ?)";

preparedStatement = connection.prepareStatement(query);

preparedStatement.setInt(1, id);

preparedStatement.setString(2, name);

preparedStatement.setString(3, date);

preparedStatement.setString(4, address);

**int** result = preparedStatement.executeUpdate();

**if**(result!=0) {

System.***out***.println(result + " no. of rows affected");

System.***out***.println("Data inserted successfully");

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Dynamic read using command line:

**package** com.te.jdbcapp;

**import** java.io.FileInputStream;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.Properties;

**public** **class** DynamicRead {

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

Connection connection = **null**;

PreparedStatement statement = **null**;

ResultSet resultSet = **null**;

**try** {

FileInputStream fileInputStream = **new** FileInputStream("db\_info.properties");

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

properties.load(fileInputStream);

//Step 1;

// Class.forName("driver");

//Step 2

connection = DriverManager.*getConnection*(properties.getProperty("dbUrl") , properties);

//Step 3

String query = "select \* from primary\_info where id=?";

statement = connection.prepareStatement(query);

//Set Values

statement.setInt(1, Integer.*parseInt*(args[0]));

resultSet = statement.executeQuery();

//Step 4

**while** (resultSet.next()) {

System.***out***.println(resultSet.getInt("id"));

System.***out***.println(resultSet.getString("name"));

System.***out***.println(resultSet.getString("date"));

System.***out***.println(resultSet.getLong("phno"));

}

} **catch** (Exception e) {

e.printStackTrace();

}

**finally** {

**if**(connection!=**null**) {

**try** {

connection.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**if**(statement!=**null**) {

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**if**(resultSet!=**null**) {

**try** {

resultSet.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

}

}

}

Try with resource Dynamic update using command line:

**package** com.te.jdbcapp;

**import** java.sql.\*;

**public** **class** TryWithResource {

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

String url = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

String query = "update primary\_info set name=? where id=?";

**try** ( Connection connection = DriverManager.*getConnection*(url);

PreparedStatement prepare = connection.prepareStatement(query);) {

// Step1

// Class.forName(properties.getProperty("driver"));

// Step 3

prepare.setString(1, args[0]);

prepare.setInt(2, Integer.*parseInt*(args[1]));

**int** result = prepare.executeUpdate();

// Step 4

**if** (result != 0) {

System.***out***.println("updated");

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

How to convert a util date to sql date:

**package** com.te.jdbcfirst;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.Scanner;

**public** **class** DateFormat {

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

**try** (Scanner scanner = **new** Scanner(System.***in***);) {

System.***out***.println("Enter any date");

String stringDate = scanner.next();

System.***out***.println("Date as string " + stringDate);

Date date = **new** SimpleDateFormat("dd.MM.yyyy").parse(stringDate);

System.***out***.println("Date format for java.util " + date);

//Here we can't import java.sql.date

java.sql.Date sqlDate = **new** java.sql.Date(date.getTime());

System.***out***.println("Date as sql format " + sqlDate );

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

In a staticInsert program convert util date to sql date.  
  
**package** com.te.jdbcfirst;

**import** java.sql.\*;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** FirstInsert {

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

Connection connection = **null**;

Statement statement = **null**;

String stringDate = "2020.02.21";

Date date= **null**;

**try** {

date = **new** SimpleDateFormat("yyyy.MM.dd").parse(stringDate);

} **catch** (ParseException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

java.sql.Date sqlDate = **new** java.sql.Date(date.getTime());

// Step 1: Load and register the Driver

String dbUrl = "jdbc:mysql://localhost:3306/ems?user=root&password=root";

String query = "Insert into emp\_info values (13, 'Vinay' ,'"+ sqlDate +"', 'Marathahalli')";

**try** {

Class.*forName*("com.mysql.jdbc.Driver"); // Step 1

// Step:2 get the connection via Driver <DriverManager>

connection = DriverManager.*getConnection*(dbUrl); // step 2

// Step:3 create a statement obj and pass the query via connection.

statement = connection.createStatement(); // step 3

// Step:4 Process the result returned by db

**int** result = statement.executeUpdate(query); // step 4

**if** (result != 0) {

System.***out***.println(result + " no of rows affected");

System.***out***.println("Data inserted succesfully");

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**try** {

**if** (connection != **null**) {

connection.close();

}

**if** (statement != **null**) {

statement.close();

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

}

DynamicInsert using date in commandline:

package com.tyss.jdbc.dynamicquery.common;

import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.text.SimpleDateFormat;

import java.util.Properties;

public class InsertUsingDynamicQuery {

public static void main(String[] args) {

Connection connection = null;

PreparedStatement prepare = null;

try {

FileInputStream fileInputStream = new FileInputStream("db\_info.properties");

Date dob = new Date(new SimpleDateFormat("dd/MM/yyyy").parse(args[2]).getTime());

Properties properties = new Properties();

properties.load(fileInputStream);

// Step1

Class.forName(properties.getProperty("driver"));

// Step2

connection= DriverManager.getConnection(properties.getProperty("dbUrl"),properties);

// Step3

String query = "insert into primary\_info values(?,?,?,?)";

prepare= connection.prepareStatement(query);

prepare.setInt(1, Integer.parseInt(args[0]));

prepare.setString(2,args[1]);

prepare.setDate(3, dob);

prepare.setLong(4, Long.parseLong(args[3]));

int result= prepare.executeUpdate();

// step 4

if(result!=0) {

System.out.println("Inserted sucessfully");

}

} catch (Exception e) {

e.printStackTrace();

}

// Step 5

finally {

try {

if(connection!=null) {

connection.close();

}

if(prepare!=null) {

prepare.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

}

Show them a maven project.