

Adv Java course

- [Before Java :: WebSite user /Visitor] Batch code : RTA0414
- [After Java :: Website Developer] mail id : netwicelearner@gmail.com
- pre-requisite : core Java topics + Common sense TD Group: netwicejavaarena (private group)
- duration : 180 sessions (Single slot)
- prerequisites : None
- Main topics :: 1. JDBC + MySQL
2. Servlet, JSP + JBoss
3. Jsp + JBoss
4. JBoss Projects + JBoss
5. HTML, java script + JBoss

Further :: core Java + advance module + OOPs + Spring framework
1x experience :: Further Topics + Intermediate [with clients] [10+]
2x experience :: 1x topics + fundamentals
3x /4x/hr :: 2x topics + spring microservices + angular /angularjs /reactJS + design [optional]

==> Since we can use Java to develop different types of Applications like standalone Apps, websites, mobile Apps, Dedicated Apps and etc., so we can call Java as programming suite, instead of just calling it as programming language.

3 Important modules of Java Programming suite

- [1] JRE module [Java Standard Edition]
- [2] JRE module [Java Enterprise Edition] [JavaEE Enterprise Edition]
- [3] JVM module [Java Micro Edition]

JRE module

 => It is base module for other modules in Java.
 => It is installable software for Java development API.
 => It is standard module to work with Java in our computer.
 So we should install this module called JRE software as minimum software in our computer in order to start our working with Java.

 >> Latest version : Java14 / Jdk14
 >> Stable version : jdk11 / Jdk11

Core Java Topics (Important)

-
- oops
- Exception handling
- Collections
- Multi-Threading
- String handling
- javafx features

==> This module is mainly given to standalone Apps and Applets.

Standalone App [1x - User App]

 ==> The App that is specific to one computer and only one user (User) at a time to operate the app is called Standalone App.

note:- The person who operates in App is called end-user (like bank employee, bank customer, website visitor and etc..)

examples for Standalone Apps

- a) Calculator App b) Java calc with main() - method c) anti-virus s/w

and etc...

Two Types of Standalone Apps

 1) GUI Application [Graphical User Interface] [GUI :- Character user interface]

 => These are not graphical Apps, So these are not user friendly.

 => Generally these Apps run via command prompt.

eg: class with main() - method with out using AWT/Swing..

TestApplication.java

 public class TestApplication {

public static void main(String args[]){

 }

class TestApplication.java

and class TestApplication

2) GUI Application / Desktop Applications [GUI :- Graphical User Interface]

 ==> These are graphical Apps and user friendly Apps.

 ==> Java Swing Frame Apps, are called standalone GUI Applications.

 ==> These Apps can be executed from Command prompt or Frame (Desktop)

eg: Calculator App , class with main() - method having Awt/Swing support

 Anti-Virus s/w and etc...

Anti/Swing based Java GUI App

 public class ArithmeticApp extends JFrame{

 }

public static void main(String arg[]){

 }

1

 //Frends.java:ArithmeticApp.java

//Frends.java:ArithmeticApp

 Anti Frame/Swing App

 value1:: 100

value2:: 200

 result:: 300

 User Learning is all about [price , what you want]
 (a) Language Learning [Core Java] [like raw material] --> wheat powder/ice powder
 (b) Technologies/WPS Learning [Adv.java] [like semifinished product]
 (c) Frameworks Learning [Spring, Hibernate, webServices] [like fully finished products]

 [like Maggie noodles, Noodles..]

IIS (Windows)

Java Enterprise Edition [old name]
 Java Enterprise Edition [new name]
 Latest Version - 7.0
 >> IIS is not a installable software - It is set of technologies like servlet,jsp,ejb and etc.
 acting as a/w specifications providing set of roles and guidelines to develop webserver (like Tomcat)
 These are developed based on IIS module supplied roles and guidelines...



IIS module is giving only Technologies like servlet, jsp, EJB, JMS, Java Mail, JPA, JTA,

MASS and etc...
 IIS --> Enterprise Java Beans [installed]
 Jsp --> Java Server Pages
 JMS --> Java Messaging Service
 JTA --> Java Transaction API

JMS --> Java Authentication and Authorization Service
 >> Using IIS module and its webServer/Application server (i.e., JMS) we can develop Enterprise Apps like webApplications and Distributed Apps/Remoteing Apps

- 1. WebApplications (WebSites)
- 2. Distributed Applications (Remoteing Applications)

Struts, Spring, JBoss Seam and etc... are frameworks and not part of Java modules (IIS,JMS,JPA). They need to be arranged separately.

Web applications/ Websites

>> It is a client-server app (two-tier App) where client is browser and server is a software App having ability to take browser generated requests, processing the requests and sending the output back to browser as response.



>> we can use Servlet/JSP Technologies of IIS module and kind JavaScripts like web technologies to gather to develop websites (web applications) in Java.

Distributed App / Remoteing App

The application who logs/services can be accessed from different types of remote clients is called Distributed Application.



In Distributed Application area, Both client and server are the software Applications, the methods/operations of Server App will be called from client software Apps like mobile Apps, ATM apps, websites and etc...
 J2EE x/w

>> We can RMI (installed), EJB (installed), CORBA (installed) and webServices (installed) to develop Distributed Apps ...



Every SW Project will have two Domain names

- [A] Functional domain name
→ Will be based on the functionalities of the project
eg:- banking domain, insurance domain, telecommunication domain and etc..
- [B] Technical domain name
→ will be decided based on the technology in which the project is developed
eg:- Java, PHP, .NET, Python, C++ and etc..

JEE module useful to develop Banking, e-commerce, retail, networking, and etc., domain projects..

JEE module techniques and less frameworks can be used to develop Banking, Insurance, Financial Services (BFS), Billing, Health Care, Traveling, Inventory, e-commerce and etc., domain projects.

JME module was used... to develop Telecommunications, Mobile Gaming, AI and etc., domain projects.

note: JDBC, Hibernate are not there to develop complete Java Apps... different types

Apps can use JDBC/Hibernate to make their Apps talking DB s/v., and to manipulate DB table records..

Explain what is software vendor, software services company and Client Organization?

Software Vendor

The company who creates and releases software to the market is called as software vendor company

- eg:- IBM/Microsoft , Apple, Oracle corp ,Apache , Google and etc..

Software services Company

These companies use the software released by the software vendor companies and develops software projects for client organizations based their requirements..

- eg:- Infosys, wipro, accenture, CTs, Vellios, TCS and etc..

note:- IBM, Google and etc., companies will act as both software vendor and software services company

for example:- IBM India develops software projects by gathering requirements from client organizations... where IBM USA acts pure software vendor company

Client Organization

It is the organization who give their requirements to s/w service company to develop software projects ..

- eg:- State Govt, central Govt, Airtel, TATA, SBI, Airbus, Indian Railways ,GlobeBank, Anexis and etc..

Some Big Client organisations running their own software companies

- eg:- UNO , Welsfergo ,IP Morgan and etc..

What is the difference b/w Software product and software service Project?

→ Software products are projects that are developed by software company and kept on open market for multiple client organizations to test and purchase..

- eg:- Tally , MSOffice, Wipro, Anti-Virus soft , Adobe Photoshop and etc..

→ Software service Project /software project are developed software service companies exclusively for one client organization by gathering requirements from that client organization..

- eg:- Citibank developed by POLARIS
Anexis Project developed by CTs
Airtel project developed by Verizon
and etc..

Different types of software companies

(A) Private limited Companies

→ Develops software product to the market

- eg:- Dross, Microsoft, UHG, Welsfergo and etc..

(B) Joint venture companies

→ Develops software projects for client organizations by gathering requirement from them..

(C) Vendor Company

→ creates and release software...

- eg:- Microsoft, google , IBM , Apache ,Oracle corp and etc..

(D) Solution Provider companies

→ Provides hardware or software solutions to improve the existing business or to create new streams of business..

- eg:- Redhat , Dropbox ,Accenture , facebook and etc..

Zoom Link: [https://transitzoom.us/inviting/Register/1UICjpu/MuJNQoSwEuf7Dk-8B3xJ_waR2-PweswvzTf4I0-\(Sun Day 13 am](https://transitzoom.us/inviting/Register/1UICjpu/MuJNQoSwEuf7Dk-8B3xJ_waR2-PweswvzTf4I0-(Sun Day 13 am)

Casualday1 :-

Ans: static block: one time executing block where as constructor is object level one-time executing block ...

When JVM loads the class ... the static block of the class will be executed only for 1 time...
In the entire App execution the JVM Loads the given class only for 1 time... so static block executes only for 1 time... we use static block to initialize static variable... we place our initializations in static block ... we established connection with DB (w/only for 1 time irrespective of no. of objects that are created...)

```
static {  
    --  
    --  
}  
  
AVM executes constructor for each object creation... we place logic to initialize non-static variables in the constructor ... If we create multiple objects for java class ... the constructor executes for multiple times... we place object level one time execution logic in the constructor... for example if want to see every object should use separate connection to interact with DB s/w... then better to write logic in the constructor...
```

```
<access modifier> <class name> {<params>}  
--  
--  
}  
  
Example  
-----
```

```
software setup n: jdk any version [1.8+], eclipse / netbeans+
```

In One Java file we can place multiple class definitions... when we compile this file

multiple .class files will be generated on 1 per each class definition basis...

==> In One Java file [source file] only one class can be taken as public class ... In this situation public class name and .java file name must match...

```
E:\Adit\Java  
|---> basics  
    |---> TestApp.java  
  
note: Only java compiler generates 0-param constructor is called default constructor.  
but the programmer placed 0-param constructor can not called as default constructor.
```

```
int a; //default is 0  
int a=1; //0 is initial value of "a"  
  
In case computer we can see multiple java  
execution simultaneously... and different  
jms can have different versions...
```

```
Class.forName("....");  
--> it is static method of predefined class java.lang.Class  
(class name it self class)  
  
|--> This method makes underlying VM to load the given java class  
dynamically at runtime... But does not create the object for the loaded  
class. So only static block of loaded class will execute... not the constructor.
```

--> If the given class is not to load then It throws ClassNotFoundException...
(checked exception) which should be either catch and handle or should be declared
to be thrown using "throws" clause...

```
eg: Class.forName("Demo");  
    Class.forName("java.util.Date");  
        we should pass fully qualified  
        class name here ... (class name with package  
        name)
```

```
Demo.java  
public class Demo {  
    static Test  
}
```

```
Test.java  
public class Test {  
    static {  
        System.out.println("Test static block");  
    }  
    // appearance can defined constructor... not the default constructor  
    public Test()  
    {  
        System.out.println("Test's Constructor");  
    }  
}
```

```
DemoTest.java  
public class DemoTest {  
    static {  
        System.out.println("TestApp: static block");  
    }  
    public void test() throws DemoException {  
        System.out.println("Test of method 'test'");  
        Test t=new Test();  
        t.print();  
        System.out.println("-----");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        System.out.println("End of main() method");  
    }  
}
```

Assignment:
(Q) When java does not import package why does it throw NoClassDefFoundException?
(Q) What is the difference b/w System.out.println() and System.out.print()

```
Demo.java  
public class Demo {  
    static {  
        System.out.println("Demo static block");  
    }  
    public void print() {  
        System.out.println("Demo's print()");  
    }  
}
```

```
DemoTest.java  
public class DemoTest {  
    static {  
        System.out.println("TestApp: static block");  
    }  
    public void test() throws DemoException {  
        System.out.println("Test of method 'test'");  
        Test t=new Test();  
        t.print();  
        System.out.println("-----");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        Class.forName("Demo");  
        System.out.println("End of main() method");  
    }  
}
```

VM attempts to load the Java class in the following situations

- [a] When we class to "java" tool having main() method to begin the execution
cmd> java TestApp

[b] In the Process of instantiation(object creation) JVM loads the class ,if it is not already loaded.
Test (Name Test);

[c] when we call static method on the class name

String str=String.valueOf(10);
Thread t=Thread.currentThread();
Class.forName("a");

[d] When we pass certain class name as the argument value of Class.forName() method.

Class.forName("javatut.Date");
[here we are explicitly asking JVM to load the certain class- java.util.Date]
forwards [makes jvm to load the given "javatut.Date" class dynamically at runtime]

(e) If we take class/interface/annotation/enum name as the parameter type or return type method ... then they will be loaded automatically.

```
public String sayHello(String helloName){  
    ...  
}  
and etc...  
notes: while loading any class... the jvm also loads all the classes of inheritance hierarchy belonging that class...
```



Can we develop Java App with main() method?

Ans] => upto java6 possible using static block

<> in java7 not possible

>> in Java 8/10 possible using static block but our app should be developed as Java8 applications.

>> Not possible from Java11 onwards..

upto Java 6 Java App with out main() method

DemoApp.java

```
public class DemoApp{  
    static{  
        int a=10;  
        System.out.println("square value::"+a*a);  
        System.exit(0);  
    }  
}
```

```
cmd> set path=C:\Program Files\Java\jdk1.6.0_31\bin  
cmd> java -version  
cmd> java DemoApp.java  
cmd> java DemoApp
```

↙ (Success)

Java app with out main() method in java7

DemoApp.java

```
public class DemoApp{  
    static{  
        int a=10;  
        System.out.println("square value::"+a*a);  
        System.exit(0);  
    }  
}
```

cmd> set path=C:\Program Files\Java\jdk1.7.0_80\bin

cmd> java -version

cmd> java DemoApp.java [success]

cmd> java DemoApp [error]

Error: Main method not found in class DemoApp, please define the main method as:

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

Developing Java app with out main() as Java8 App in Java11/10 version

//DemoApp.java

```
public abstract class DemoApp extends java.util.application.Application{  
    static{  
        int a=10;  
        System.out.println("square ::"+a*a);  
        System.exit(0);  
    }  
}
```

}[/DemoApp]

```
cmd> set path=C:\Program Files\Java\jdk1.8.0_31\bin  
cmd> java -version  
cmd> java DemoApp.java [success]
```

There is no java8 in jdk before java version...

```
cmd> java DemoApp [error]
```

output:
square :: 100

Developing Java App with out main() from Java11 onwards

Not possible .. because Java is removed from jdk from Java11 onwards...

/DemoApp.java

```
public abstract class DemoApp extends java.util.application.Application{  
    static{  
        int a=10;  
        System.out.println("square ::"+a*a);  
        System.exit(0);  
    }  
}
```

}[/DemoApp]

```
cmd> set path=C:\Program Files\Java\jdk-11.0.3\bin  
cmd> java -version  
cmd> java DemoApp [error]
```

DemoApp.java:3: error: package java.util.application does not exist

public abstract class DemoApp extends java.util.application.Application

^

3 errors

Stable Java requires ::
↳ java11, Java11

What is the difference between System.out.println() and System.err.println()?

Ex-1] I supports output redirection where as .out() does not support

output redirection... by default...

/System.out.java

```
public class SystemOut{  
    {  
        public static void main(String args[]){  
            System.out.println("comes");  
            System.err.println("comes");  
        }  
    }  
}
```

[/System.out.java]

/DemoApp.java

```
/DemoApp.java > output.txt  
cmd> java SystemOut  
comes  
comes  
cmd> cat output.txt  
comes  
comes
```

>>> cat output.txt for normal debugging /output messages

>>> java -Xlog:err for catch[] blocks related exception messages..

API (Application Programming Interface)

====
It is no way related to Java meaning, It is used having English meaning like base, platform and etc.
=> API is base the programmers to develop software App ... every programming language, s/he technology and framework provides API.
=> In "C" language API means set of functions that comes in the form of "header files" (.h file).
=> In "C++" language API means set of functions and classes that comes in the form "header files".
=> In "Java" language API means set of classes, interfaces, enums, annotations which comes in the form of packages..

3 types of APIs

- a) Pre-defined APIs / Built-in APIs
 - > Will come along with software
- b) User-defined APIs
 - > Developed by the Programmers
- c) Third party APIs
 - > Given by Third party vendors like Java poor, softwareTree, Apache and etc.

part1 --- Programmer/Developer

part2 --- The s/he vendor company who is giving software language/technology/framework

Examples Java pre-defined APIs

utility api --> java.util and its sub-pkgs.
reflection api --> java.lang.reflect and its sub-pkgs.
networking api --> java.net and its sub-pkgs.
jdbc api [Database] --> java.sql, java.sql and their sub-pkgs.
swing api [GUI] --> javax.swing and its sub-pkgs.
international api [International] --> java.util and its sub-pkgs
and etc..

=> In Application development we can use all 3 types of APIs..

=> set of APIs together is called library ... and these libraries in Java will come as jar files...

[jar files = Libraries= api = set of packages = classes+interfaces+enums + annotations]

[jar file]-----java archive [java based zip / jar file]
-----compressed / zip/jar format

In java .jar files will be used in different angles, they are

=> as zip file or rar file [To combine multiple files into single file].
=> To represent pre-defined>User-defined/Third party api's/Library.

--> Entire Java language pre-defined API comes in the form of rt.jar file.

=> represents JRE driver API's.



=> JDBC driver is bridge s/he between Java App and DB s/w..
which capable of converting Java calls to DB calls and vice-versa.

e.g.: oracle/7/8/9jar [Oracle JDBC driver s/w]

=> Represents ready to release standalone Project [will release app/Project as executable jar file].

e.g.: tomcat-every-jar

=> Represents ready to deploy/host Java web application [website].

here we work war file [a flavor of jar file]

War is a jar files representing Java based website..

war --> web application archive.

=> represents EJB-Cores (Distributed Apps)

and etc...

note: Every API/Library comes in the form of jar file.. but we can not say every jar file is an API/Library because jar files even represent JDBC driver s/w, standalone projects , web sites and etc...

=> To create our own jar file .. we can use the JDK supplied tool called "[jar]"

(available in c:\java\bin directory.)

To create jar file

E:\Advise\Java\Demo>jar cvf n1.jar
---->--> current directory
---->--> parent directory
E:\Advise\Java\Demo>jar cf n1.jar -
--> create archive file
--> specify archive file name
E:\Advise\Java\Demo>jar cf n1.jar -
added manifest
adding: file1 = 19K (data 103 deflated 33%)
adding: file2 = 50K (data 32 deflated 38%)

-->--> manifest directory

-->--> parent directory

all the content [both files and subfolders] will be added n1.jar file having compressed size!

To see the content jar file -----juse windo or whatever

E:\Advise\Java\Demo>jar tf n1.jar
META-INF/
META-INF/MANIFEST.MF
file1
file2
E:\Advise\Java\Demo>jar tf n1.jar
0 Wed Jul 15 17:19:58 IST 2009 META-INF/
66 Wed Jul 15 17:23:30 IST 2009 META-INF/MANIFEST.MF
18 Wed Jul 15 17:23:26 IST 2009 file1
52 Wed Jul 15 17:23:38 IST 2009 file2

-----table of content...

To add new file to the existing jar file [Updation of the jar file]

E:\Advise\Java\Demo>jar uvf n1.jar 123.txt
adding: 123.txt(file = 452) (data= 413 deflated 9%) s--> update the archive
E:\Advise\Java\Demo>jar tf n1.jar
META-INF/
META-INF/MANIFEST.MF
file1
file2
123.txt

-->--> extract from archive

To extract all the content form jar file

E:\Advise\Java\Demo>jar xf n1.jar
created: META-INF/
extracted: META-INF/MANIFEST.MF
extracted: file1
extracted: file2
extracted: 123.txt
notes: you can not delete specific file or files or
extracted: 123.txt
extracted: 123.txt

-->--> extract from archive

what is the difference between PATH and CLASSPATH?

PATH

=> It is DOS/Windows command... So it can be used while working all software that are running in windows/DOS computer... This command is not java specific command.
=> In order execute .exe , .jar , .cmd files certain location from any location of that computer with out copying files... we need to add the original location of these files to PATH env. variable...

batch (.bat) file is useful to create single command by combining multiple commands.

run.bat

date cmdc run
time cmdv run.bat
ver
dir
here "run" "run.bat" acts as single command having multiple commands execution internally.

Problems:

E:\a\java>
|---->PATHDemo
|---->run.bat

run.bat

date
time
ver
dir

E:\a\java>PATHDemo>run (success)

>>> run.bat (success)

E:\> run [error]

=> that command or file name is not (or)
= > not is not an internal or external command or executable program or batch file (new)

E:\a\java> run [error]

C:\> run [error]

notes : we able execute run.bat file only from its original location... not from locations...

Solution1: (Not Recommended)

copy run.bat file other location ... where u want to execute... this is not recommended... better when copy file to other locations... memory will be wasted for the same file..

Solution2: (Recommended)

Add the original location of run.bat file (E:\a\java\PATHDemo) to PATH env. variable...

This pc --> properties --> advanced settings --> app. variables --> user/ps variables
variable name :: PATH
value :: E:\a\java\PATHDemo;c:\windows;

-->ok -->ok -->ok

c:\windows>

user env. variables are specific to currently logged in windows user... where as

system env. variables visible to all the windows user of computer...

E:\a\java>run (success)

E:\a\java>PATHDemo>run (success)

All java tools like javac, java, jar and etc... are .exe files of c:\windows\bin directory.In order to use them... true location of computer... based compilation and execution of java code... we need add c:\windows\bin directory PATH env. variable...

variable name :: PATH

value :: C:\Program Files\Java\jdk1.8.0_111\bin;c:\windows;

Explain diff ways of reading inputs from user??

CLASSPATH

↳ Java can't find class based env. variable... so it can be used only in Java env., not in all other places
↳ If Java App uses user-defined apis or third party apis [other than jdk apis] then those apis related files or directories must be added to CLASSPATH env. variable... In order to make java tools like javac, java and etc... recognizing and using those apis...

Problems:

User-defined API [WishAPI]

F:\Java\javase\1.8.0_111\lib\ext
|---WishMessageGenerator.java
|---WishMessageGenerator.class
|---WishMessageGenerator.class

WishMessageGenerator.java

```
package com.ctk.basics;  
public class WishMessageGenerator {  
  
    public String sayHello(String user){  
        return "Good Morning "+user;  
    }  
}
```

F:\Java\javase\1.8.0_111\bin> -> WishMessageGenerator.java

Main Application [Which wants to use WishAPI]

E:\Java\javase\1.8.0_111\bin> MainApplication.java [Application]

MainApplication.java

```
import com.ctk.basics.WishMessageGenerator;  
public class MainApplication {  
  
    public static void main(String args[]){  
        WishMessageGenerator generator=new WishMessageGenerator();  
        String result=generator.sayHello("naveen");  
        System.out.println(result);  
    }  
}
```

E:\Java\javase\1.8.0_111\bin> MainApplication.java [error]
E:\Java\javase\1.8.0_111\bin> java MainApplication [error]
E:\Java\javase\1.8.0_111\bin> java MainApplication [success]

Conclusion: To run Java in our computer we need add <java_home>\bin directory to PATH env. variable... similarly to make Java App working with user-defined, third-party apis they are need to add their locations to CLASSPATH env. variable to make Java tools like javac, java and etc... recognizing and using package and its content [com.ctk.basics and its classes] from that location...

This PC ---> properties ---> advanced settings ---> env. variables ---> user/env variables--->

```
variable name: CLASSPATH  
value :: F:\Java\javase\1.8.0_111\bin  
->ok->ok...
```

E:\Java\javase\1.8.0_111\bin> java MainApplication.java [success]

E:\Java\javase\1.8.0_111\bin> java MainApplication [success]

notes: we can use user defined java packages/api's location to PATH...env. variable... because Java tools like javac, java always looks in CLASSPATH env. variable... not in PATH env. variable...

Preparing jar file - representing user-defined api

F:\Java\javase\1.8.0_111\bin> jarsigner -keystore wishapi.jar - alias wishapi.jar (representing user-defined api)
added manifest
adding com\ctk\basics\WishMessageGenerator.class (size=1024) (deflated 37%)
adding com\ctk\basics\WishMessageGenerator.class (size=1024) (deflated 37%)
adding com\ctk\basics\WishMessageGenerator.class (size=1024) (deflated 37%)
adding com\ctk\basics\WishMessageGenerator.class (size=504) (deflated 37%)
adding WishMessageGenerator.java (size=1024) (deflated 25%)

If user-defined or third party apis are there in the form of jar files... then we need add jar files names and locations to CLASSPATH env. variable... Inorder make java tools using them...

```
variable name :: CLASSPATH  
value :: F:\Java\javase\1.8.0_111\bin\wishapi.jar;old values;
```

E:\Java\javase\1.8.0_111\bin> MainApplication [success]

E:\Java\javase\1.8.0_111\bin> MainApplication [success]

notes: Up to level we can also copy jar files that representing user-defined or third party apis to <java_home>\lib\lib\ext folder... [we can not place directories here]

eg:- copy F:\Java\javase\1.8.0_111\bin\wishapi.jar file to <java_home>\lib\lib\ext folder...

note: The content "ext" folder will be used by "ExtensionClassLoader" ... But ExtensionClassLoader is removed from "java9" onwards. So there is "ext" folder in the JRE installations from java9...
Q) What is the difference b/w adding values to CLASSPATH and adding values to ext folder ?

Important points on env. variables

- a) Env. variables are not case-sensitive
- b) Values set to env. variables must be separated with ":"
- c) ":" symbols added to Env. variables represent "current directory", "parent directory" respectively.
- d) It is recommended to add new values to "env. variables" at the beginning...
- e) Values set to env. variables from cmd prompt will remain temporary... i.e we will loose those once cmd prompt closed... whereas the values kept in "This PC" env. variables will remain permanent.
- f) Modifications done in env. variables will not reflect in old cmd prompts... so we must open new cmd prompts... and etc.

What is persistence?
It is the process of saving and managing data for long time... for
this we need to use File, DB, DBMS and etc.

Name of Persistence

- The Data stored in Variables, objects, collections, arrays and etc... are stack-based memory
- all the data stored in RAM is called temporary memory or the RAM memory of the App will be vanished once the application terminates. **RAM** is temporary memory. It can't store App data across the multiple executions of same App or diff Apps.

To overcome this problem... we need to write **JDBC API** Code to **Persistance Memory Services** like **Filesystems, DB, DBMS, persistence and etc...** by taking the support from **DB** or **etc...**



Imp Technologies related to Persistence

- [x] Persistence
- [x] Persistence Store
- [x] Persistence Model
- [x] Persistence operation
- [x] Persistence logic
- [x] Persistence Technologies/Frameworks
- (Data Source / Technologies / Frameworks)

Persistence

→ it is the process of saving and managing data for long time.

Persistence store

→ the place where the data can be saved and managed for long time
→ it is also Data Storage Unit / Data storage software.

- [x] File - BLOB, Google drive, One Drive, Cloud, Dropbox and etc...
- [x] Zone
- [x] Database

Persistent Data

→ The data of Persistence store is called Persistent data...
eg: File content, DB table Records...

Persistence operations

Read, update, delete and select operations performed on the persistent data are called Persistence operations. These are called as CRUD(CREATE/READ/UPDATE/DELETE) operations...

CRUD/CRIO	SCRD	Events
C—create [insert]	SCRD	Events: Session start project file
U—update [modify]	U—updated [Read]	Running App, Timer/Booking ,
C—Create	C—Create	U Commerce and etc... should deal with CRUD Operations...
U—Update [edit]	U—updated [Read]	
D—Delete [remove]	D—Delete	

Persistence logic

→ it is the high level code using which we perform persistence operations on persistent data...
eg: Job code, Alternative code, UI elements code and etc...

Persistence Technology/Frameworks

→ it is the technology or framework using which we develop Persistence logic... These are the most popular Persistence frameworks.

- eg: JPA (Technology), Hibernate (Framework), Spring JPA (Framework), Spring CRIO (Framework), Spring Data (Framework), etc...

Persistence Technology/Framework is given for developing persistence logic to perform persistence operations on the persistent data of Persistence store.

New App → JPA → Persistence store → The file that software and hardware independent are called file like XML files, entities, and files and etc...
, .xml, .do, .class, .db and etc... are not file...)

New App → JPA → RDBMS DB like Oracle, MySQL, PostgreSQL and etc...

Limitations with files on Persistence stores

- [x] No Security
- [x] No support for huge amount of data...
- [x] No SQL Support
- [x] It is very complex to retrieve data by applying multiple conditions.
- [x] No SQL Support, So data redundancy/delegation problem may occur
- [x] Performing update and delete operations is very complex.



[x] Can not handle relationships... [the FK column]

[x] Comparing and merging of data is very complex...

and etc...

To overcome these, Persistence are **DB like** as the Persistence stores...

Conclusion

If the App is small size App... and generating little amount of data to persist. Then for this the Persistence stores...
eg: File, Database, Google Sheets, Excel, Google Sheets and etc...

If the App is medium scale or larger scale App...generating huge amounts of data to persist.
Then go for DB like as Persistence stores...
eg: Banking Apps, E-Commerce Apps, Financial Mangers Apps, and etc...

NoSQL

1—employees with 20 details
2—employees with 10 details
3—employees with 5 details
4—employees with 2 details

DB like
Employee having 30 calls...
(DB rows are not pre-constructed
Data doesn't exist)

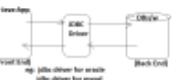
To overcome these problems, we get NoSQL DB like... which allows to store data in different formats like JSON, XML and etc... but not as rows and cols...

- [x] MongoDB, Hazelcast and etc...

New App → MongoDB, Hazelcast and etc... → NoSQL DB like
MongoDB, Hazelcast and etc...

notes on DB: all databases are not good for bulk HIGH DB like... they need separate their own apps...

DB Driver
DB Driver API's bridge between Java App and DB w/o requiring Java calls [Instructions] i.e.
DB calls and DB calls to Java calls ...



JDBC drivers act as bridge b/w Operating Systems and physical Devices e.g. Printer drivers, Second Drivers and etc...



JDBC drivers
Open Database connectivity

When all Java Apps are using JDBC Driver to interact with DB q/b, why Java App use JDBC Driver separately JDBC drivers?

- a) JDBC drivers are global "C" language language supports of pointers... but Java does not support pointers.
- b) JDBC drivers are not portable logic to non-pointer orientation logic and using.

- c) JDBC drivers are developed in "C" language which is Platform dependent... so pairing JDBC driver with Platform independent Java App like Java App Portability i.e. Platform independent behaviour.

To overcome all these problems... Java App are using Java Based JDBC driver q/b.

We use Java JDBC drivers from All vendors

- a) JDBC group (Oracle, MySQL, IBM, Oracle corp and etc...) [Best]
- b) All vendors
- c) Third party Vendors

mailto:suresh.suresh@gmail.com

What is the diff b/w Programming language and Use Technology?

Programming language

- i) It is directly installable q/b acting as raw material by providing basic features that are required in Application development.
- ii) Java, C, C++, Python, Java Script, C# and etc...
- iii) All the programming languages are machine independent.
- iv) Programming languages are base create q/b technologies... [B/w q/b... Operating systems, Device Drivers, Tools, Frameworks and etc...]

q/b [Java, C, C++, etc...], tools, frameworks, python and etc...

Java based language

Java, Java Derby, JBoss, Go, Jenkins...

All these languages gives class files after using their compilers, but those class files can be executed using same jvm.

Software technology

- i) It is a software spec [Specification] that gives bunch of rules and guidelines to the form of q/b.
 - ii) All the software technologies follows by writing one or another programming languages.
- q/b [Java, C, C++, Python, Java Script, C# and etc...]
- Implementation q/b are not installable, but the q/b technology based implementation q/b are installable or executable... Working with these implementations q/b is nothing but working with q/b technologies.
- a) JDBC Driver
i) JDBC Driver is a software spec [Specification] that gives bunch of rules and guidelines to the form of JDBC API [Java API, Java q/b] and plug it to develop JDBC driver q/b's.
 - b) JDBC Driver
i) JDBC Driver is not installable just JDBC based JDBC driver q/b's are installable [like jar]... Working JDBC drivers is nothing but working JDBC Driver.

eg: JDBC, hibernate, JPA, JEE, Java mail, JPA, JMS and etc...

Offshore Phase of Delivery/Project Development (With respect to developer)

- (i) Development : Coding | ↳ API's DR's
- (ii) Testing : (I) Checking the code | ↳ API's DR's
- (iii) Release & Production : Adding project to client organization and client using that project | ↳ DR's
- (iv) Maintenance/Support : Fixing the bugs as assigned | ↳ DR's

DR's have to:
1. — implement classes implementing various interfaces/api's and released to the market as jar file.

→ Every JDBC driver will be identified with its JDBC driver class name i.e. `com.mysql.jdbc.Driver`, `com.oracle.jdbc.Driver`, `com.ibm.jdbc.Driver` and `com.jtds.jdbc.Driver` etc in our Java code by just using its JDBC driver class name.

→ This java class implements JDBC driver by implementing `java.sql.Driver` interface or indirectly it called `JDBC driver class name`.

In `applicationContext.xml`:

```
<code><jdbc:driver>com.mysql.jdbc.Driver</jdbc:driver> </code> → JDBC driver class name  
plugin name: JDBC driver class name  
JDBC driver class name implementing java.sql.Driver
```

code any Java code and can articulate any JDBC driver's behavior by creating JDBC driver class or by creating object for JDBC driver class.

App-Java beans to interact with Oracle DB's:

↳ understand JDBC API's file then interact and add changes
→ Assume JDBC driver's name from Internet and add changes

↳ Choose appropriate JDBC driver driver definition → Binding of JDBC driver class
(i.e.)
`oracle.jdbc.driver.OracleDriver` → JDBC driver name or use JDBC driver class name in the code
↳ JDBC driver class name

→ use JDBC driver's name to develop the persistence logic (CRUD operations)

App-Java Client to interact with MySQL DB's:

↳ download MySQL JDBC API from Internet and add changes

→ Assume JDBC driver's name from Internet and add changes
↳ Choose appropriate JDBC driver driver definition → Binding of JDBC driver class
`com.mysql.jdbc.Driver` → JDBC driver class name

→ use MySQL JDBC driver's name to develop the persistence logic (CRUD operations)

[MySQL-Connector-Java-8.0.24.jar]

<https://dev.mysql.com/doc/relnotes/jdbc/en/> → for MySQL JDBC driver's jar file

↳ Application/Java/Code/ProjectName/WEB-INF/lib → place MySQL.jar file

[MySQL-8.0.24.jar (translation)]

Java compiler is given below! to generate byte code(.class) from source code (.java)

[Build-in with JDK]

↳ Java Developers are given the generate code (.java) from byte code (.class), we need to understand how to convert byte code to source code.

To use Java Decompiler → go to Decompiler zone → choose and upload either .class file or jar file having class file and get the source code.

```
java.sql.Driver {  
    implements  
    oracle.jdbc.driver.OracleDriver()  
}  
In applicationContext.xml
```

```
java.sql.Driver {  
    implements  
    com.mysql.jdbc.Driver()  
}  
In MySQL-Connector-Jar-Source.jar
```

>> JDBC_driver is to bridge between Java App and DBs
 >> the JDBC_driver implements interface Driver –> means bunch of classes implementing this interface can implement Driver interface –> JDBC_driver class name must implement java.sql.Driver interface
 >> required constructor: java.sql.DriverManager (Jdbc_driver's name) –> By using this driver name in our app we can activate and use jdbc_driver type.
 >> The java class that implements java.sql.Driver[] is called jdbc_driver class and it is unique class name for every jdbc_driver type...
 In jdbcDriver.java [Jdbc_driver's driver's name]
 =>> public driver class contains bunch of methods to do different things like:
 plug name: class name implementing java.sql.Driver[]
 In general construct for javaDriver.jar [Jdbc_driver's driver's name]
 =>> public driver class name = com.mkyong.jdbc.JdbcDriver
 (A) plug name: class name implementing java.sql.Driver[]
 (B) implements: com.mkyong.jdbc.JdbcDriver[]
 (C) implements: java.sql.Driver()
 (D) implements: com.mkyong.jdbc.JdbcDriver()
 What is the difference [A] [B] [C], driver class... and driver type is [D]?
 Ans) >> JDBC_driver's type is bridge between Java App and DBs ... and comes in the form of jar file
 =>> Every [Driver's type] is identified , can be activated and used by using [Driver driver class name]
 =>> The java class that implements java.sql.Driver[] is called [Driver], driver class.


In DriverManagerService

- => In the registration of many Java App lots of services will be shared/internally, like DatabaseCollector service and DriverManager.
- => DatabaseCollector service is useful for delivery the objects in Java App.
- => DriverManager Service is useful to manage the drivers in Java App.

The DatabaseCollector service is useful for delivery the objects in Java App using System.gc(), Runtime.gc() methods.

The DriverManager Service is useful for delivery the objects in Java App using various methods: java.sql.DriverManager class...

notes: In order to use jdbc_driver type in our Java App, it need be registered/registered with DriverManager service ... with the support of java.sql.DriverManager class...

Register JDBC_driver type with DriverManager service is nothing, keeping JDBC_driver class object in DriverManager service...

a) Registering once JDBC_driver type with DriverManager service

o Create JDBC_driver object
 o DatabaseCollector.registerDriver(new JDBC_driver());
 o JDBC_driver class object with DriverManager service.

DriverManager.registerDriver(driver);
 static method

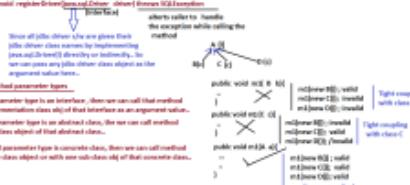
b) Registering JDBC_driver class object with DriverManager service:

o If JDBC_driver object to talk with only one DBs then there are no need to register multiple JDBC_driver's type, with DriverManager service. (eg: Involving memory leak problems of cause same)

o If JDBC_driver objects to talk with more than one DBs type then we need to register multiple JDBC_driver's type with DriverManager service. (eg: Involving memory leak because of two different banks)

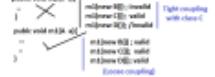
Signature of registerDriver() method of DriverManager class:

```
public static void registerDriver(DatabaseDriver driver) throws IOException {
```



3. Imp. notes on method parameter types

- [a] If java method parameter type is an interface, then we can't overload method having same interface type as well as all that interface as an argument value.
- [b] If the method parameter type is abstract, then we can't overload having one sub-class object of that abstract class.
- [c] If the java method parameter type is concrete class, then we can't overload either with concrete class object or with one sub-class obj of that concrete class.



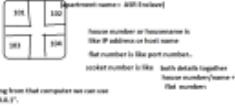
Port number or "socket number"

>> In our computer port 1024-65535 are given to connect hardware devices. In our computer
eg:- keyboard port, USB port, printer port, mouse port, Joystick port and etc.
Operating system gives software ports 32768-65535 to applications services, software services
netcat port 4444, Oracle port 1521, MySQL port 3306 etc. These ports are available.
In that 3 to 65535 ports reserved for OS related services (Windows Services and the
remaining software ports are given for externally installed apps (DBS to 65535 like oracle,
mysql, tomcat and etc.)

external_use	default	portnumber
	1024	=> what is port number check?
mysql	3306	=> ports less than 1024. Preempt [oracle] default port no: 1521
postgresql	5432	=> ports server default port = 5432.
tomcat	8080	=> ports range [any number]
and etc..		=> any number or can choose below 1024 to 65535

Socket numbers [In a connection of port number]

socket number : <= > databases/host_name + port_no.
oracle port number : 1521.55.77.61:1521 [ip]
oracle socket number : 1521.55.77.61:1521 [oracle] 30:1521
socket number is called communication port/pool, because using socket number
we can connect to any service running on any computer.



notes: To write a connection string from that computer we can use

"localhost" or "127.0.0.1".

oracle port no: 1521

current computer oracle socket number : localhost:1521

127.0.0.1:1521
like oracle:1521:1521

Oracle Database TNS Listener based jobs driver job for oracle is called "Oracle Listener" jobs driver

It is connecting to the host jar file along with Oracle installation. like sqlldrca.jar

like sqlldrca.jar

Oracle Listener driver details are

>> jobs_driver class name :: oracle.jobs.OracleDriver Job

>> jobs url set :: jdbc:oracle:thin:@(host):(port):(service_name):tcp://(host):(port)

host name or
host ip address of
oracle db which this
connection is connected
connection
where oracle_db
is running

example url:
1) jdbc:oracle:thin:@(host):(port):(service_name) (for local db oracle)
or
2) jdbc:oracle:thin:@(host):(port):(service_name) (for local db oracle)

3) jdbc:oracle:thin:@(host):(port):(service_name) (for remote db oracle)
(host)
4) jdbc:oracle:thin:@(host):(port):(service_name) (for remote db oracle)
(host)

jar file :
differs in how oracle handles the network
difference 1: oracle 11g handles 1111 as (for local db oracle)
difference 2: oracle 11g handles 1111 as (for remote db oracle)
spfile11g (oracle 11g)
spfile11g (oracle 11g+), where remote oracle DB is running

protocol
connection

connection is bunch of rules followed by two parties , who want participate in
communication.

>> there are two types of protocols

a) instance protocols
connection based rules to get communication b/w two physical computers ..
eg:- TCP/IP, UDP and etc.

b) Application protocols

Uses set of rules to get communication b/w two software or software applications..
eg:- http or https (to get communication b/w browser & web)
difference 1: to get communication rule, uses App and oracle db (using JDBC driver s/he support)

difference 2: to get communication rule, uses App and mysql db (s/he using JDBC driver s/he support)

and etc...

In application protocol
jobs to make protocol
rule is auto generated
in TCP/IP
TCP is made protocol
IP is auto protocol

note: Application Protocols runs over (on the top of) network protocols



Write a JDBC App [Java app with JDBC] to establish the connection
and make DB querying made this driver

step[1] Keep the following JDBC code ready

```
createDBFileForAnyServices([DB1], [oracle 11g]  
oracle.jdbc.driver.OracleDriver [DB1])  
catalogplus [Database]
```

step[2] make sure that Oracle DB & its related services are running properly.

```
Get the search -> services.msc -> services -> search for  
ORACLEDB1[Oracle's logical DB name] or service name in comment.  
SQL> select * from global_name;  
XE_10GR2 [orcl] 10GR2 [db]
```

step[3] Getting the driver by details [oracle thin driver detail]

```
refer driver details [oracle thin driver detail]  
...for all services [and]  
step[4] Add driver [as needed per file to choose or copy]  
class.getResourceAsStream() based on vendor java services.  
[only update if need to]
```

```
note: If Java app uses any [as file content] [the packages/bases]  
variable name = CLASSINPUT  
value = C:\JDeveloper\app\src\main\java\com\oracle\thin\DriverManager.java  
or -> -> ->
```

step[5] Develop the Application.
All the methods in System class and
DriverManager class are static methods.

```
DriverManager  
|-----+  
|-----getDriver()  
|-----+-----getDB1  
|-----+-----getDBs  
|-----+-----getDBList()  
+-----Connection
```

Connection.java

```
{Connection.java  
import java.sql.*; //JDBC API}
```

public class Connection{

```
[  
    public static void main(String args){//throws exception;  
        //register oracle thin driver with DriverManager service  
        //it creates a connection pool for oracle thin driver  
        //oracle jobs driver OracleDriver driver means oracle jobs driver connection  
        //register jobs driver  
        DriverManager.registerDriver(jobs);  
        static method  
    }  
} //Establish the connection with DB's url  
//DriverManager.getDriver(url).getconnection("url", "username", "password");  
Connection con=DriverManager.getConnection("jdbc:oracle:@localhost:1521",  
                                         "system", "manager");  
//Establish the connection  
//DriverManager.getDriver(url).getconnection("url", "username", "password");  
System.out.println("Connection is established");
```

step[6] compile and execute the Application..

```
cd -> java Connection.java  
cd -> java Connection
```

Connection con=DriverManager.getDriver("file:/c:/oracle/jdbc/oracle10j.zip");
static method
↓

```
[DriverManager] -----| method called on DriverManager class performs  
[a] Register "file:/oracle/jdbc" url of jdbc zip. It makes the driver manager service to pick up  
the registered "oracle thin driver" from Driver Manager service.  
[b] Establishes a connection with Oracle DB using Oracle's logical DB "orcl" which is running on  
[c] port number having logical DB "orcl" on the current current computer and establishes  
the connection type "oracle" logical DB orcl and java App. In that Process the it collects  
all the information about the connection and returns it.  
[d] returns "Oracle connection object" [it implementation class obj of java.sql.Connection]] back  
to Java App representing connection with DB type .. we are referring that JDBC connection object  
using javax.sql.Connection ref variable [con]
```

Instead of calling `DriverManager.registerDriver(jdbcDriver)` with `DriverManager` service, we can write single line of code for the same that is `DriverManager.registerDriver(driverClass)`.

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

In above line, there can be any driver registered with `DriverManager` service because we are just loading JDBC driver class here.

And, `Class.forName()` method creates java instance for JDBC driver class dynamically at run time. In this process, it also loads the JDBC driver class resources and that is why we can see that JDBC driver class has its own resources like properties file that object calls `DriverManager.registerDriver(registerDriver())` method on.

In the static block of oracle.jdbc.driver.OracleDriver class—

```
static {
    try {
        if(DriverManager == null) {
            DriverManager = new OracleDriver();
            DriverManager.registerDriver(OracleDriver());
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

static block of com.mysql.jdbc.Driver class—

```
static {
    try {
        if(DriverManager == null) {
            DriverManager = new OracleDriver();
            DriverManager.registerDriver(OracleDriver());
        }
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

In all other driver classes of different JDBC driver API, we will be having this kind of static block. So we can register all JDBC drivers with JDBC Manager service by just loading their JDBC driver classes... with the support of `Class.forName()` method...

we can write any code using JDBC driver class that makes static JDBC driver

code getting executed... for register JDBC driver...
eg:- `new oracle.jdbc.driver.OracleDriver().registerDriver(oracle.jdbc.OracleDriver());`

eg:- `new oracle.jdbc.driver.OracleDriver().registerDriver();`

In above two examples, before creating objects, JDBC driver class loads given JDBC driver class and its static block executing register logic to register JDBC driver with JDBC Manager service. And the logically created objects for JDBC driver classes will be created... So these are very good techniques.

-ORACLE Technology latest version is 14.1 -> 12c

or create local service = 12c

or create this driver linked service = apilevel12jar

or Java linked service = 12c

driven by JDBC driver class name or create binding driver class... so no need to load JDBC driver class directly or indirectly... with the help of JDBC Manager service.

eg:- create 12c.jar file and add it to "Driver after 3 sessions and it does not require static loading of driver class..."

-ORACLE recommends "Auto-loading of driver class" feature is introduced... As we need to load JDBC driver class directly or indirectly... we just add jar file to class path and the driver class will automatically loaded...

eg:- `DriverManager.registerDriver(new oracle.jdbc.OracleDriver());` we are giving based on JDBC, so they support auto-loading JDBC driver class... So there is no need of writing `Class.forName()`

<http://www.java2s.com/Code/jdbc/DownloadDriver12jar.html>

(This is get .apilevel12jar file)

If you are using JDBC driver in linked descriptor, then we must write `Class.forName("oracle.jdbc.driver.OracleDriver")` code in our App.

If oracle 12c... (If we have .apilevel12.jar file in addition then writing `Class.forName("oracle.jdbc.driver.OracleDriver")` code in our App is optional..

How "Auto Driver Loading of JDBC driver class feature" is working from JDBC API methods?

Q: If we call to `DriverManager.getConnectioon()`... I find the static block of `DriverManager` class executing... This static block having logic to search for `META-INF/services/java.sql.Driver` file in all jar files that are added to `C:\AWARD\lib` and it finds in one jar file like `apilevel12.jar` file... it collects JDBC driver class name from that Java and Oracle API and loads that class internally by using reflection API.

apilevel12.jar file file is having `java.sql.Driver` file in `META-INF/services` folder

as driver below... as JDBC support auto-loading driver value

`java.sql.Driver [file]`

`oracle.jdbc.OracleDriver...`

apilevel12.jar is given based on JDBC and it does not maintain `java.sql.Driver` file...

So it does self-loading of JDBC driver class...

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306","root","manager");

DriverManager.getConnection(...); returns `Driver` object - and returning object by returning `getConnnection()` method.

Q `DriverManager` is an interface, then how can we say `DriverManager.getConnnection()`??

Ans(1) `Driver` can only be an object of `[java.sql.DriverManager]` as we know the interface can not be instantiated. `Driver` can object is the object of underlying JDBC driver's supplied Java code. `DriverManager` is an interface which is implemented by `[java.sql.DriverManager]`. `DriverManager` is the parent class of `Driver` class. `Driver` can object `conn` is `Driver` driver's child object. `Driver` driver's child is `Driver` driver's class and `Driver` is the `DR` class. so we never refer that `Driver` can object using its original actual class ref variable, we always refer by using the common `Connection` ref variable as there is heavy coupling..

System.out.println("Driver can object class name : "+conn.getClass());

In Oracle this driver type is `OracleDriver` for oracle db the `Driver` can object class name is:

oracle.jdbc.driver.OracleDriver

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

we can use "new" to refer "[Driver can object]" when `Driver` drives like `type2`

`drivers = new OracleDriver();` instead `drivers`, `OracleDriver` and etc. **Java only coupling**

`FileInputStream conn=DriverManager.getConnection();` ✗

here can not be "use" to refer other `Driver` driver's related `Driver` can object. **Type coupling**

notes: `DriverManager.getConnection()` internally creating and returning the `Connection` class object. does not concern itself based on `Driver` class who we are using. As `Driver` class is the parent class of `Driver` class, `Driver` class can implement `Connection` class object, so we can refer `Driver` can object given by `DriverManager.getConnection()` method using `[java.sql.Connection]` ref variable [as]

Ans(2) `java.util.List` means it is the object of `java.util.List` class that implements `[java.util.Listable]`.

`Classmate` object means it is the object of `Classmate` class that implements `[java.lang.Comparable]`.

`ArrayList` object means it is the object of `ArrayList` class that implements `[java.util.List]`.

`Driver` can object means it is the object of `java.sql.Driver` class that implements `[java.sql.Driver]`.

notes: Project has a class having `String name` with `Interface` name `Driver`. It is not the object of `Interface`. It is the object of `[Driver]` class implementing that `Interface`.....

Ans(3) `Signatures of DriverManager.getConnection()`

`public static Connection getConnection(String url,
String username,
String password);` Multi派生

this method does not return `[java.sql.Connection]` object because interface can not be instantiated. i.e. this method is returning `[DriverManager]` connection object which is the concrete value object as `Driver` can object.

Using statements are modified after its type.

a) If `pass` method return type is an `interface`... then that method returns one impl class object can not be instantiated. i.e. this method is returning `[DriverManager]` connection object which is the concrete value object as `Driver` can object.

b) If `pass` method return type is an `abstract class`... then that method returns one sub class object of that abstract class.

c) If `pass` method return type is a concrete class... then that method can return either that concrete class object or one of its sub class object.

public "A" "B" "C" in J (String type)

↳ Hypo assumed "A"
return new A();
↳ Hypo assumed "B"
return new B();
else if type equals ("C")
else
return new C();

A B
Implements
Implements

X(Y)
Y(Z)
Z(D)

A extends ("A")
B extends ("B")
C extends ("C")

D extends ("D")

Java only coupling -
Driver can object
object through common interface
not concrete with one becoming
the main target

→ JDBC Statement object works by Java API and DB's API to convert SQL queries as inputs to DB's API from Java API and to get SQL query results back to Java API from DB's API.

To create JDBC Statement object

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

Statement[] method internally creates and returns the implementation class of Java's Statement[] as JDBC Statement[]. Statement[] is a subclass of Statement[].

According Oracle Specification there are 4 types of SQL Queries:

- 1) DML Queries (Data Manipulation Language) → Insert, Update, Delete SQL Queries
- 2) DMS Queries (Data Manipulation Language) → Create Table, Drop Table, Alter Table and etc.
- 3) T-SQL Queries (Transact-SQL Language) → Comment, Authorize, Asymmetric and etc.
- 4) TCO Queries (Transact-Code Language) → Comment, Authorize, Asymmetric and etc.

According Java JDBC, we have two types of SQL queries:

- 1) Select SQL queries - JDBC Statement
- 2) These queries give bunch of records [0 or more records]
- 3) eg: JDBC Statement → executeQuery()

II) PreparedStatement SQL Queries (DBMS, JDBC, TSQL, SQL)

→ These queries return numeric value (mostly EMR, quoted representing no-of-rows affected).

→ SQL queries can be set and update values (max response=2000);

- 1) rows updated
- 2) insert into table values(10, 'Vimal', 'Vipul', 47, 400,
- 3) rows & new inserted

→ SQL executeQuery() on JDBC Statement object to send execute SQL query to DB server and to get JDBC Statement object having bunch of records...

ResultSet result = statement.executeQuery("select * from dept");


executeQuery() → ResultSet object

→ SQL executeUpdate() on JDBC Statement object to send execute SQL query to DB server and to get JDBC Statement object having bunch of records...

ResultSet result = statement.executeUpdate("insert into dept values(10, 'Vimal', 'Vipul', 47, 400);");


executeUpdate() → int

If the above query deletes 0 records then it returns 0 value , if the

above does not delete the records then it returns 1.

signature of executeUpdate():

public int executeUpdate(String sql) throws SQLException;



It is checked exception

Handling different problems...

Write a Jdbc App to get all records from db table "student"

step 1) create db table in oracle having name "student" as shown below

SQL> create table student(id number(3), name varchar(20),addr varchar(30),age number(2,2));

step 2) insert records into db table

SQL> insert into student values(3,'Hari', 'Mysore', 25.88);

1 row created.

SQL> insert into student values(2,'Vidya', 'Mysore', 25.88);

1 row created.

SQL> insert into student values(1,'Rajesh', 'Mysore', 25.88);

1 row created.

SQL> commit;

step 3) Develop the JDBC App using oracle thin driver [Oracle java supplied type]

[oracle.jdbc.driver.OracleDriver]

Select first few rows from student

100 | 1 | Rajesh | Mysore | 25.88 | 01 |

101 | 2 | Vidya | Mysore | 25.88 | 02 |

102 | 3 | Hari | Mysore | 25.88 | 03 |

BIG> > Before First Record

AE> After Last Record

RE> Recordset is ResultSet and contains all records in db table.

Resultset will have 3 record index.

>>> When Resultset object is created, by default cursor resides in "WTF" position

cursor[0] -- moves the cursor in Resultset object in forward direction...i.e from

current position to next position... if the next position is record then

it returns null else cursor[1] returns false.

while(cursor[0] != false){

 System.out.println("id : "+cursor[0]+", name : "+cursor[1]+", age : "+cursor[2]+", address : "+cursor[3]);

}

100 | 1 | Rajesh | Mysore | 25.88 | 01 |

101 | 2 | Vidya | Mysore | 25.88 | 02 |

102 | 3 | Hari | Mysore | 25.88 | 03 |

we can call rs.getmethod() methods either with col index or with col name to get col values

From the record of Resultset object...

rs.getmethod() are like rs.getint(), rs.getString(), rs.getDouble(), rs.getDate() and etc...

>>> If db table [any from 1 to 3] can not read unconnected data from DB table...

or >>> If db table has no primary key then it can't read all columns as listing values... by

rs.getmethod() method either by passing column or col index...

>>> It is recommended to close cursor after end of their utilization... that is reverse of their opening...

or >>> If db table has no primary key then it can't read all columns as listing values... by

rs.getmethod() method class the "ResultSet" object and does not allow to reuse the cursor

in the Resultset object and does not allow to process the records in the Resultset object

create method class the "Statement" object and does not allow to send execute

or >>> If db table has no primary key then it can't read all columns as listing values... by

rs.getmethod() method class the "Statement" object and does not support further

interactions with db table...

Assignment:-
a) Write a JDBC App to get all records for Dept Obj table
b) Write a JDBC App to get Empno,ename,job,Sal
col values from emp obj table.

To get default db table to use

step 1) connect to oracle db using JDBC program

create connection object

step 2) search for work.apf file in Oracle bin directory

step 3) copy and paste the content of work.apf on SQL prompt from line number 34 to 40 and

SQL or select * from student where snum>100 and snum<200;

class for performing
using scanner class...

If built-in stream of java
→
a) System.in → points to keyboard (standard input device)
b) System.out → points to console (standard output device)
c) System.err → points to console (standard output device)

System.in	
100 <= snum <	input : 100,200
100 <= snum <	[] []
100 <= snum <	100,200
100 <= snum <	100

snum = read input : 100,200
100 <= snum < data : 100,200



value

→ All 3

Scanner(s) → Reader

StringTokenizer → Scanner

Scanner → StringTokenizer

StringTokenizer → String
String → getting essence of the input
BooleanValue → Boolean
False ==> false
True ==> true
→
we are getting reverse of input

System.out.println("Flag?" "results not found");
"Students whose snum>=sdata" and snum<=edata" details are displayed";
if Flag is false
→ No output
if Flag is true
for i of "Students whose snum>=sdata" and snum<=edata" details are displayed";
→
return;

Where jno does not present, why does it throw NullPointerexception?

Data itself, or Data &
dataReady(); → throws NullPointerexception

→ When we invoke method or variable on the reference variable that holds/points to "null" value
then "NullPointerException" will be raised.
→ here "dataReady" method/variable is not yet created in 3.0+ pointers. It is
used using "engulf" mechanism. Meaning method/variable is defined on the ref variable that is
pointing to "null" value.

notes: In Java ref variables are capable of holding addresses of the objects, So pointers
concept is not required in Java.

JDBC API's Links

connection

<https://github.com/VarunSaraf/Jdbc/blob/main/src/main/java/JDBC>

a) Write a JDBC App to get Student details based on given initial characters of student name

b) Write a JDBC App to get Group details based on given Gid range.

Q1 Write a JDBC App to get employee names from emp divisible based on place

3 three jobs order by job

select empno,ename,job,dept from emp where job in ('CLERK','SALESMAN','MANAGER') order by job desc;

exercising 3 jobs these exercises ->
converting them to appearance -> displaying them as required for
klipper app -> writing programs -> hence exercise is
non-technical & we can not expect non-technical SOF
type inputs.

note - SQL supports , DB table names, db col names are not variables, but the db col values are
case sensitive:-



If Resultset obj is holding specific col values of DB table,
then we should print col indices in the order they are
stored in Resultset obj(hence col ordering in the SQL Query).
ie we should not be col order in the DB table.

Assignment :-

- (a) Write job app to get student details based on given 3 city names.
- (b) Write job app to get Emp details based on given empno.

What is difference Editor and IDE?
 → Editors provide development environment (Typing env.) Let them do not provide multiple options that are required in App development - testing, debugging, execution, packaging, release, deployment, editing, etc.
 → IDE (Integrated Development Env.) is a single UI who's where development, testing, debugging, execution, packaging, release, deployment and etc... are possible.
 a) Eclipse IDE (open source (commercial), releases, IDE, Java, C/C++, C/C++/Android Application developed, Java, JavaEE, and etc...)
 Popular Java IDE - Eclipse IDE
 IDEs (IDE) - Eclipse IDE
 notes: Eclipse is good. But Eclipse with XSD plugin is more good.
 problem: It is patchy or additional effort required for it. That can be avoided by installing the necessary additional components to existing XSDs.

Eclipse
 version:
 type : IDE for Java
 name : Eclipse IDE - Java EE Developer, 2020-12 (Updated)
 Vendor : Eclipse
 Open source
 Features : Eclipse IDE Java IDE for Java (and Java-based development)
 Eclipse IDE (The off type of Apps development) 128-bit or 64-bit
 % download Eclipse : download or setup file or zip file
<https://www.eclipse.org/g把手/Downloads/Eclipse.php>
 Mac-MacBook-Pro:~/Downloads/Eclipse-2020-12/eclipse-
 macos-x86_64_v2020-12/eclipse-IDE-Mac.zip
 To install Eclipse :
 1. click [Extract] or select [Install].
 2. use eclipse.exe from extraction to launch.
 3. use desktop icon to launch.

Write Java App to get Employee details based on the given Employee number?
 (develop using Eclipse IDE)
 Q1) < select employee_name(job_id) from emp where employee_id;
 collect from editor

Procedure
 step[1]: make sure Eclipse is installed in u/c computer ... and launches it by choosing `eclipse/eclipse` folder
 (The folder where Apps/
 Projects will come)
 [E:\WorkSpace\JavaProject\113111]

step[2]: Create Java Project using Eclipse having name JDBCPProj.
 File menu → new → Project → Java Project → name: JDBCPProj → next → next → ...

step[3]: Add ejb.jar[V1.1/1.1.1.jar file to the buildpath/builpath of the above Project.]
 right click on project → build path → [select jar files / Libraries, and in configuration tab → Libraries tab → add external jar file] ejb.jar → click finish
 note: If you want to use ejb.jar, application.jar needs to be added separately.

step[4]: creates new package in "src" folder of the project . having name "com.ejbe".
 right click on "src" folder → new or (ctrl + n) → package → name: com.ejbe.java
 step[5]: develop the little Java by adding code 1 method in "com.ejbe.java".
 right click on "com.ejbe.java" plug → class → class name: SelectField
 select main method check box → ...
 Ctrl+Shift+/ → to increase decrease font size.
 Ctrl+Shift+I → Import all code.
 Ctrl+Shift+R → Refactor all code and for doing inheritance.
 import ejb.Session : glass.EJB ; (ctrl + space)
 ejb.create : ejb.create(ejb.Session session) with message
 ejb.create(ejb.Session session, Object bidContent) as ejb.create(session, content);
 ejb.create(ejb.Session session, Object bidContent) as ejb.create(session, content);



ctrl + space: it is more seconds than use ejb.create(); to process the resultset.
 ctrl + space: it is 1 second than use ejb.create(); to process the resultset

Step[6]: run the application.
 use F10/F11
 [ctrl + shift + f11] → run
 right click on java file [where main() is there] → run as → Java App...
 (ctrl + shift + f11) → run button

Assignments :
 1) Write a JDBCP App to get student details based given student name
 2) Write java App that gives employee details who are having highest salary

Write a JBoss App to Imp. count from emp(oracle)



what count() from emp:

what Resultset obj contains shape one value either 0 or positive number -> a resultset object can have 0 or more rows -> an empty object, for we can directly call count() method to get the Resultset object to its name containing [Resultset] or what's count() method.

Important statements of ResultSet Processing:

a) If the Resultset object is going to have 0 or more records then use while(rs.next()) to process the Resultset object.

else used. * from emp where id='11111' gives 0 or more records

b) If the Resultset object is going to have 0 or 1 record then use if(rs.next()) to process the Resultset object.

else select count(*) from emp... gives 1 record always

What the diff b/w calling toString() method and calling printStackTrace() method on any exception obj?

Any Throwable called on exception-object gives just exception class name and message for the exception that is raised.

>> printStackTrace() called on the exception-object gives the elaborated details of the exception that is raised. If exception is raised, the stack memory of method execution will be filtered with messages, to print those messages on the console ...then the printStackTrace() ..

note: If any exception will not be used to display program output. Then for programmers, they can use printStackTrace() method of println-messages will be rendered through Web Front end like Interceptor or on webfragments...

What is the need JBoss and JBoss app development ...where we can read and execute SQL Queries to DB or from SQL Prompt?

a) End-users are more interested example, as they do not know SQL... interacting SQL queries from SQL prompt is not possible for them... As they expect one query as front App to interact with DB, he would be manipulating DB like the data ... based on the inputs given by end-user...

b) If any developer wants to use SQL Query to interact with DB, then he needs to write some code to DB like data... which can be missed.... So avoid this one-front App is required to allow end-user to perform restricted operations on DB like...

To develop such-front End App using jax... we need to take support of JBoss... directly or indirectly...



b) To get the source code of printed code

for JBoss

c) HttpServlet :: To get source code one random class/Interface/ enum /annotation [query]

Indicates user input

Executing non-select SQL Queries
--> (1) insert, update, delete and (2) create table, alter table, drop table...
--> (3) comment, rollback, commit... -> queries are called non-select SQL Queries.
--> execute SQL query (statement object) value representing n'th records that are effected after executing the SQL query.
--> broader executeUpdate (use Statement object) to send and execute Non-select Query in DB.
DB API
public int executeUpdate(String query) throws SQLException
--> while working with non-select SQL Queries there is no need of dealing with ResultSet object.

QWrite a Jdbc App to insert record into EMP table by collecting inputs from enduser?

SQL> insert into student values(101,'Vishal','1997-01-01')
--> The inserted SQL Query that goes to DB will be from Java App executes in DB if we are writing individual code on DB side...

make the code N.I if it deals with SQLException object we can catch user-friendly error
Message by using SQL error code that are stored in SQLException object... These error codes are specific to each DB like...

```
com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException  
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:941)  
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:896)  
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:882)  
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:869)  
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:852)  
System.out.println("cannot insert duplicate in PK column val");  
else if((err.getErrorCode())>20000)  
System.out.println("cannot insert values more than column size");  
else if((err.getErrorCode())>1000 && err.getErrorCode()<=999)  
System.out.println("SQL Query syntax problem");  
else  
System.out.println("unknown jdb problem");
```

Make alter table student add primary key();
Table altered.

Assignment:-
Q) Write a JDBC App to insert values to employees table out of emp db table?

SQL> insert into employees values(104,'Vishal',20000);

Write a JDBC App to delete records from Student db table based on the given city (Address)?

```
SQL> delete from student where city='Mumbai';  
System.out.println("records deleted");  
  
--> again Jdbc [file L1/L2] we have single executeUpdate() method to send and execute non-select SQL queries in DB if we  
--> execute [public void] statement.executeUpdate(query) throws SQLException  
--> executeUpdate [public int] signature --> public long executeUpdate(String query) throws SQLException  
--> journal file name [Large Database applications where we can manipulate DB source of records]
```

Assignment:-

Write JDBC App to delete student details based on given student number?

Using delete SQL Query we can delete complete record(s) ... it was can't delete specific row values from the records, for that we need to use update query to update specific row values.

Q)Write a JDBC App to update student address, name, age with new values

Java code for update operation:

```
String update = "update student set name=? , address=? , age=? where id=?";  
Statement stmt = con.createStatement();  
stmt.executeUpdate(update, new int[]{1,2,3,4});
```

Assignment:- Write a JDBC App to increase employee salary by 20% if the employee is

DEPARTMENT :- IT

SQL update query will be like this (put ? for value): update John SET salary=10000;

Assignment:- Write JDBC App to add 10.0 avg for students who received less than 40.0 avg.

Q) Can we execute both select and non-select SQL query by having single method on JDBC Statement object?

JDBC provides using `executeQuery()` and `executeUpdate()` methods.

Signature :- public ResultSet executeQuery(String SQLStatement)

call the executed SQL query's result SQL query items in return.
Issue :- we can't read and separate `getResultSet()` method to
the `executeQuery()` method.
call the executed SQL query's non-select SQL query items in return.
Issue :- we can't read and separate `getUpdateCount()` method to
the `executeQuery()` method.

[Download file and Application](#)

Java code for executeQuery() :

```
if(pf.executeUpdate("select * from student")){  
    ResultSet rs = pf.executeQuery("select * from student");  
    while(rs.next()) {  
        System.out.println(rs.getString("name"));  
    }  
}  
else {  
    System.out.println("No rows found");  
    int count = pf.executeUpdate("insert into student values('John', 'IT', 30, 50000)");  
    System.out.println(count + " rows inserted");  
}
```

working with executeQuery() is not good practice because it makes us to do with 2 method calls in order to complete the task.
So, we can't use executeQuery() and executeUpdate() methods alone to deal with single method call.

select query :- executeQuery() but execute() or getResultSet()

non-select query :- executeUpdate() (and execute() with getUpdateCount())

Assignment:- write a JDBC App that prints highest salary employee details

Assignment:- write a JDBC App that prints highest salary employee details

Login App Development

↳ Should Perform Authentication i.e given `username,password` must be matched with the `username,password` stored in already available in DB table.
↳ User can login only if App ... is functioning to DB like.

Step 1: Create a table having `username,password` fields

MSD: create table account(username varchar(255) primary key, password varchar(255));

Table created.

MSD: insert into account values('root','root');

1 row inserted.

MSD: insert into account values('admin','admin');

1 row inserted.

MSD: insert into account values('user','user');

1 row inserted.

MSD: commit;

Step 2: decide the best sql query to perform this Authentication.

MSD: select username,password from account where username='root' and password='root';

ORACLE:

MSD: select username,password from account where username='root' and password='root';

ORACLE:

① (would succeed)

Step 3: Develop the application.



What is SQL Injection (Hacking Technique) and how does it raises?

Ans: making special SQL injection part of the SQL query by sending them along with the normal query to change the SQL query behaviour this application.

behavior - Related SQL injection Prevention

→ Hackers and jokers use this technique to get into other accounts with out username or valid password.

Example:

enter username: root -- (correct username)

enter password: root -- (wrong password)

result:... Valid Credentials | SQL injection Prevention

Types of JDBC Statement obj:

- a) Statement obj: `Implementation class obj of java.sql.Statement[]` [Working with this obj so far]
- b) PreparedStatement obj: `Implementation class obj of java.sql.PreparedStatement[]`
- c) CallableStatement obj: `Implementation class obj of java.sql.CallableStatement[]`



Operations of Statement objects

- =====
- (A) SQL Injection many value:
 - (B) SQL Injection few value:
 - (C) SQL Injection few value involving variables is very complex as we need to convert Java Inputs to SQL Inputs as required in the SQL query
 - (D) Does not allow to take parameters (?) in the SQL query
 - (E) We can execute same SQL query many times after with same inputs or with diff inputs.

- (F) Inserting date values in SQL complex ... [see how date values in pattern that underlying

create date pattern: dd-MMM-yy eg: 25-OCT-99
insert date pattern: ? eg: ????-??-??

(G) Inserting Large Object(s) this is not possible

Note: To solve the above problems ... take the support `PreparedStatement` objects..

Once SQL query goes to DB side - the DB engine of DB will perform the 3 operations on the DB, query



parse SQL

query will be splitted into tokens and the query syntax will be verified

execute

The parsed query will be executed in the DB side

fetch

The output of executed query will be fetched back to Java App.

multiple working with railway ticket reservation or bus ticket reservation application the insert SQL query will be executed for multiple times to insert different passenger details.

multiple working like game score the same "insert SQL query" will be executed for multiple times for game.

Working with PreparedStatement object to insert 3x passenger details in a day (Railway Ticketing)

a) 3x times same insert SQL query
1) 3x times same insert SQL query with input values : $1,00,00,000 * 3.001 \text{ sec} = 3000 \text{ sec}$
2) 3x times same insert SQL query will be processed $3000/\text{sec} = 1,00,00,000 * 0.001 \text{ sec} = 3000 \text{ sec}$

b) 3x times same insert SQL query will be executed in DB side : $3000/\text{sec} = 3000 \text{ sec}$

c) 3x times same insert SQL query output is fetched : $3000/\text{sec} * 0.001 \text{ sec} = 3000 \text{ sec}$

In the above table performing [A][B] operations on same query for multiple times is uncessary [3000 sec is wasted] and perform [C][D] operations on same query for multiple times is necessary, but while working with Simple Statement obj we can't avoid [A][B] operations happening for multiple times.

We can overcome the problem by using pre-compiled SQL query with the support of JDBC

PreparedStatement object

pre-compiled SQL query

→ The SQL query that goes to DB is not the same (parameterized SQL query irrespective of the number of times it is called) as in case of pre-compiled SQL query

→ JDBC PreparedStatement object represents/holds Pre-compiled SQL query (i.e. we can reuse multiple pre-compiled SQL query and we can results from that query execution with support of PreparedStatement object).

→ This pre-compiled query can be with inputs or can be with out inputs

→ This pre-compiled query is the static query with / without inputs / can be dynamic query.
insert into student values(10,'Vishal'); -- static query with inputs
select * from student; -- static query with out inputs
insert into student values(11,12,13); -- dynamic query with params

→ Single Statement obj deals with only static SQL query with or with-out inputs

→ PreparedStatement obj deals with both static and dynamic SQL queries

For passenger railway ticket reservation, using PreparedStatement object

a) Insert SQL query goes to DB side with parameter
1) $1 * 3.001 = 3.003 \text{ sec}$
b) Insert SQL query will be parsed in DB side
only for 1 time
c) 3x times same insert SQL query will be executed in DB side for 3x times
setting diff input values to query params
d) 3x times same insert SQL query result will be fetched and $3.003 \text{ sec} * 3 = 9.009 \text{ sec}$

→ Only [C][D] operations are happening on the same query only for 1 time.

→ No parsing overhead as we can use memory storage of time while working with PreparedStatement object.

Procedure to work with PreparedStatement obj: To execute same SQL query for multiple times

step ① create SQL query having parameters [placeholders] & parameters [] with diff inputs

String query="Insert into student values(?,?,?)";

2.2.4

step ② To make the SQL query as pre-compiled SQL query and return PreparedStatement obj representing that pre-compiled SQL query.

PreparedStatement pstmt=connection.prepareStatement(query);

This method takes the given SQL query, sends to DB i.e., makes the SQL query as pre-compiled SQL query and returns PreparedStatement object holding that pre-compiled SQL query.

step ③ use setInt(..., ...) methods on PreparedStatement obj to set the values to the query present of PreparedStatement object

pstmt.setInt(1,10);
pstmt.setInt(2,"Vishal");
pstmt.setInt(3,100);

step④ execute the pre-compiled SQL query

int small=pstmt.executeUpdate();

step⑤ process the results

if(pstmt!=null){
 if(small>0){
 System.out.println("Record inserted");
 }
 else
 System.out.println("Record not inserted");
}

notice: To run same query execution for multiple times with diff inputs, repeat 3.4.3 steps...

step ⑥ free up the resources

con.close();

While a Java App to run "Insert record student details to DB table"

→ Insert SQL query should be executed for "n" times to insert "n" student

→ need to prepare using PreparedStatement object .

note: Because of SingleStatement obj, when we query for one record it takes 3 sec to make 3000 SQL query for another record, so it makes 3000 obj to compile 3000 query, so we can't control SQL query will definitely execute.

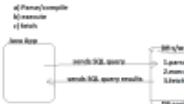
In case of PreparedStatement query compilation takes place irrespective of whether it will be executed or not.

Conclusion on Simple Statement,
PreparedStatement obj

a) In the entire Application, If there is a need of executing SQL query only for 1 time that is with out any input values then prefer simple Statement object
syntax: select * from student;

b) If there is a need of executing query with inputs for 1 or more times , without inputs for more times, to insert date values, then go for PreparedStatement object.

Given SQL query goes to DB which - the DB engine of DB who performs the 5 operations on the SQL query



Java Listener

query will be splitted into tokens and the query syntax will be verified

execute

The parsed query will be executed in the DB server

both

The output of executed query will be fetched back to execute java App.

multiple working with railway ticket reservation or bus ticket reservation application the insert SQL query will be executed for multiple times to insert different passenger details.

insert operation for same query like same "select SQL query" will be executed for multiple with different parameters

Working with PreparedStatement object to insert 3x passenger details in the RailwayTicketing

- a) 3x times same insert SQL query
b) Insert SQL query will be inserted in DB like with input values = 1,00,00,000 * 3.000 secs = 30000 secs
c) 3x times same insert SQL query will be inserted in DB like with input values = 1,00,00,000 * 3.000 secs = 30000 secs

- d) 3x times same insert SQL query will be executed in DB like = 1,00,00,000 * 3.000secs = 30000 secs
e) 3x times same insert SQL query output is treated = 3,00,00,000 * 3.000secs = 90000 secs = 60000 sec

In the above table performing [a],[d] combinations on same query for multiple times is unnecessary [30,000 secs are wasted] and perform [b],[e] combinations on same query for multiple times is necessary ,but while working with Single Statement obj we can avoid [a],[d] operations happening for multiple times.

Single Statement obj can be used [a],[b] operations happening for multiple times.

We can overcome the the problem by using pre-compiled SQL query with the support of JDBC PreparedStatement object.

preparedStatement SQL query

→ The SQL query that goes to the file and becomes prepared to get SQL query irrespective of multiple times its execution or not called pre-compiled query or PreparedStatement.

→ JDBC PreparedStatement object represents holds the compiled SQL query i.e we can execute pre-compiled SQL query and we can result from that query execution with support of PreparedStatement object.

To use this pre-compiled query we can set inputs or can be static inputs

This pre-compiled query can use static query with/without inputs or can be dynamic query.

Insert little student values(10, "Ravi", 21) → static query with inputs

select 3 from student → static query with no inputs

Insert little student values(1, 1, 1, 1) → dynamic query with parent

→ Single Statement obj deals with only static SQL query with or with out inputs

→ PreparedStatement obj deals with both static and dynamic SQL queries

for passenger railways ticket reservation using PreparedStatement object

- a) Insert SQL query goes to DB like with parent() 1 * 3sec = 3.000 secs
b) Insert SQL query will be parent in DB like 2 * 3.000 sec = 6.000 sec
c) If the Pre-compiled Insert SQL query will be inserted in DB for 3 times
d) The Pre-compiled Insert SQL query will be inserted in DB for 10 times for 10 times
e) The Pre-compiled Insert SQL query will be inserted in DB for 10 times for 10 times
f) Insert SQL query will be parent = 1.00,00,000 * 3.000 sec = 30000 sec
g) If 3 times same Insert SQL query result will be fetched out = 1.00,00,000 * 3.000 sec = 30000 sec
Total time = 200000 sec

multiple SQL operations are happening on the same query only for 3 times.
not for multiple times. i.e additional average of time while working PreparedStatement object.

Procedure to work with PreparedStatement obj: To execute same SQL query for multiple times
with diff inputs

step1) Create SQL query having powers {placeholder} in parent []

String query="Insert into student values(?, ?, ?);"

3, 3, 3

(step2) make the SQL query as pre-compiled SQL query and return PreparedStatement obj representing that pre-compiled SQL query.

PreparedStatement p=Statement.prepareStatement(query);



This method takes the given SQL query, sends to DB which creates the SQL statement and returns the PreparedStatement obj holding the pre-compiled SQL query.

Note: Instead of Statement.prepareStatement(), where SQL query for execution there it involves DB obj to complete SQL query to effectively execute.

Instead of PreparedStatement query compilation takes place irrespective of whatever it will be executed or not.

step3) use setfield...() methods on PreparedStatement obj to set the values to the query params of PreparedStatement obj

PreparedStatement p=Statement.prepareStatement(query);
p.setInt(1,100)
p.setString(2,"Ravi")
p.setInt(3,21)

step4) execute the pre-compiled SQL query

p.executeUpdate();

step5) process the results

{results=0
if(result>0){
System.out.println("Record inserted");}
else{
System.out.println("Record not inserted");}}

notes: To repeat query execution for multiple times with diff inputs. Repeat 3,4,5 steps.

step6) close jdbi obj

p.close();

connection.close();

Notes: a) JDBC does not accept "?" instead of values to DB side?

→ There Insert SQL query should be executed for "?" instead of Insert student

details by pre- using PreparedStatement object

p.executeUpdate();

b) It is recommended to take SQL queries in top of class as defining constant variable values for every executing and modifying of query if needed , with out searching where that SQL query is available in the entire program.

private static final String STUDENT_INSERT_QUERY="INSERT INTO STUDENT VALUES(?,?)";

constant value is better to define and generally that constant value will be for upper case letters.

c) We can also take query having as single constant variable value as {shown above} while working with PreparedStatement object hence the question comes that what is the difference between them? These local variables are not visible outside the main() or class level

Assignment : write java app to insert multiple customer details to Customer db table

by collecting them together. Jobs are {rec_name,add,bilnk}

note: p.executeUpdate(); will take care converting given query param values as required for SQL query... so programs need not to do that conversion.

Login-App development using PreparedStatement object

File.....(Execution)DB

NV

10

TU

What is no SQL injection Problem with PreparedStatement obj and why this Problem is there with Simple Statement obj?



Simple Statement obj makes DB obj to compile SQL query with inputs .So for the special SQL constructions like → SQL comment part (/* */). In query compilation and will be as broken SQL because it is a single statement which is not able to handle the continuation of SQL query... This changes Query and App behaviour. So SQL injection Problem is related.

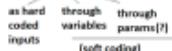
PreparedStatement obj makes DB obj to compile SQL query with inputs .So for the special SQL constructions like → SQL comment part (/* */). In query compilation and will be as broken SQL because it is a single statement which is not able to handle the continuation of SQL query... This changes Query and App behaviour. So SQL injection Problem is related.

1) → java file → java → → JDBC
2) → JDBC → DB

↳ Input
as
↓
Input
↳ Input
↳ SQL

>>The Pre-compiled Query PreparedStatement object can be a

- >select or non-select query
- >static or dynamic query
- >query with out inputs or with inputs



>>The parameters (?) of SQL query for PreparedStatement object can represent only inputs .. i.e they can not represent SQL key words , table names, cols names..

eg1: select * from student where sno=>? and sno<=? [valid]

eg2: select * from ? where sno=>? [invalid]

eg3: select ? from Student where ?=>? and ?<=? [invalid]

eg4: select * ? ? ? sno=>? [invalid]

The advantages of PreparedStatement obj:

- a) Allows to work pre-compiled SQL Query
- b) Can represent both static and dynamic SQL queries with params(?) or with hard coded/soft coded inputs
- c) Does not raise SQL injection Problem
- d) No need of converting input values as required for the SQL query
- e) Suitable for executing same query for multiple either with same inputs or inputs
- f) Suitable for insert Date values..to db table cols by collecting them from end user in different formats
- g) Suitable for inserting Large objects..(files) to Db table cols.

What is the difference SimpleStatement object and PreparedStatement?

SimpleStatement obj

- (a) Does not hold any pre-compiled query
- (b) Can be used only to execute static SQL queries
- (c) May raise SQL Injection Problem
- (d) Input values must be converted as required for SQL query
- (e) does not param(?) in the query
- (f) Not suitable to execute same SQL query for multiple times either with same inputs or with inputs ..
- (g) not suitable for date values insertion
- (h) not suitable for Large obj(files) insertions
- (i)* One Simple Statement obj can be used to execute multiple SQL queries one by one.. before it does not hold query , So it can be used to execute multiple queries one by one

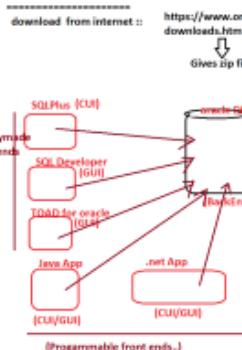
PreparedStatement obj

- (a) holds one pre-compiled SQL query
- (b) Can be used to execute both static dynamic SQL queries
- (c) does not raise
- (d) not required to convert , the setXXX(,;) methods will take care of conversion
- (e) allows ...
- (f) suitable
- (g) suitable
- (h) suitable
- (i)* Every PreparedStatement obj represents one pre-compiled SQL query, so that object should be used only to execute that query..i.e to execute multiple SQL queries, we need multiple PreparedStatement objects..

>>Java/.net/php Developers in realtime do not work with SQL prompts directly..(cmd prompts).. They prefer using GUI Java DB tools to make operations easy.. To work with these tool strong SQL knowledge is not required..

eg: SQL developer for Oracle given Oracle corp (separate installation upto 11g-Xe)
MySQL SQL workbench given by MySQL DB s/w itself
SQL yog for mysql [separate installation]
Toad for oracle (separate)
Toad for mysql
PgAdmin for PostgreSQL [will come with PostgreSQL Db s/w installation]

SQL developer for Oracle



Procedure to create DB table with records in Oracle using SQL Developer tool

step1) install and launch SQL developer

step2) Create Connection with Oracle DB s/w by giving username,password details..

new Connection [+] --> name: con1 -->submit username,password -->Test-->connect

step3) create Db table

expand: con1 --> right click tables -->new table -->

step4) perform all CURD operations on db table..

mysql
 type :: Db s/w
 vendor :: Devx /Sun Ms /Oracle corp
 open source
 default port no.: 3306
 allows to create logical DBs
 default logical DBs are **world**, **sys**, **schema**
 default username :: root
 password :: root (should be chosen during the installation)

version :: 8.x

url :: <https://www.mysql.com/downloads/>

mysql installation videos:

https://youtu.be/GfHrgjg_3Eg

MySQL Installation gives MySQL workbench as the built-in GUI DB tool

Procedure to create logical DB having DB table with records.

step1] make sure that MySQL DB s/w is installed and it is running mode

step2] create logical DB and use that logical DB..

mysql> create database ntaj414db;

Query OK, 1 row affected (0.16 sec)

mysql> use ntaj414db;

Database changed

step3] create db table student having records...

mysql> create table student(jno int(10) primary key, name varchar(15), add varchar(15), eng float);

Query OK, 0 rows affected (2.63 sec)

mysql> insert into student values(456, 'Yashvi', 'Hyd', 78.99);

Query OK, 1 row affected (0.10 sec)

mysql> commit;

Query OK, 0 rows affected (0.00 sec)

Procedure to create logical DB having DB table with records.. using MySQL workbench

step1] make sure that MySQL DB s/w is installed and it is running mode

step2] launch MySQL workbench and create connection...

Start button --> pgs --> MySQL --> MySQL workbench --> create connection (+) --> submit

details like connection name:: con1 --> ... password:: root --> test connection and ok..

step2] create logical DB and use that logical DB..

use icon --> name:: ntaj414db

step3] create db table with records...

expand logical DB ntaj414db --> tables --> Right click -->

enter the db table name --> enter the col names --> ...

--> apply --> ...

expand logical DB (ntaj414db) --> right click tables --> select all rows -->

insert values in the col boxes --> apply --> ...

--> every DB s/w needs separate type4 JDBC driver s/w's..

--> Devx supplied type4 mechanism based JDBC driver s/w for MySQL is called Connector/J

JDBC Driver

Connector/J JDBC driver details (will not come along with MySQL installation)

driver classname :: com.mysql.cj.jdbc.Driver (or) com.mysql.jdbc.Driver
 url :: jdbc:mysql://</logicalDB name> (for Local MySQL DB s/w)
 jdbc:mysql://<hostname>:<port no>/<logicalDB name> (for remote MySQL DB s/w)

host name or ipaddress machine
 and, port no of MySQL DB s/w where
 it is running

jar file :: mysql-connector-java-8.0.21.jar (supports auto-loading of driver class name)

(collect from internet --> mavenrepository.com)
<https://mvnrepository.com/artifact/mysql/mysql-connector-jar/8.0.21>

Procedure to make Java App interacting MySQL DB s/w

step1] keep any old JDBC App ready...

step2] add Connector/J JDBC driver jar file to build path/classpath..

step3] Modify the following details in the Application...

(a) driver class name
 (b) JDBC URL (c) DB Username (d) DB Password

//register JDBC driver s/w

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

/establish the connection

Connection con=DriverManager.getConnection("jdbc:mysql:////ntaj414db","root","root");

(or)

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ntaj414db","root","root");

When DB s/w is changed for JDBC App the following

JDBC properties will change

a) driver class name (optional)

b) JDBC URL

c) DB Username d) DB Pwd e) JDBC driver jar file..

note:: The class names of JDBC objects will change based on the JDBC Driver s/w and DB s/w we use ,So we never refer them with real class names related reference variables.. we always refer them with JDBC API supplied common interface reference variables .. to achieve loose coupling or run time polymorphism..

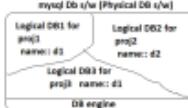


UAT :: User Acceptance Test
 also called sanity Test/Beta Test

mysql/postgreSQL can store huge amount of data..

oracle can store huge amount of data..

find out limitations of MySQL DB s/w and benefits of Oracle DB s/w?



Procedure to create logical DB having DB table with records.. using MySQL workbench

step1] make sure that MySQL DB s/w is installed and it is running mode

step2] launch MySQL workbench and create connection...

Start button --> pgs --> MySQL --> MySQL workbench --> create connection (+) --> submit

details like connection name:: con1 --> ... password:: root --> test connection and ok..

step2] create logical DB and use that logical DB..

use icon --> name:: ntaj414db

step3] create db table with records...

expand logical DB ntaj414db --> tables --> Right click -->

enter the db table name --> enter the col names --> ...

--> apply --> ...

expand logical DB (ntaj414db) --> right click tables --> select all rows -->

insert values in the col boxes --> apply --> ...

--> every DB s/w needs separate type4 JDBC driver s/w's..

--> Devx supplied type4 mechanism based JDBC driver s/w for MySQL is called Connector/J

JDBC Driver

Connector/J JDBC driver details (will not come along with MySQL installation)

driver classname :: com.mysql.cj.jdbc.Driver (or) com.mysql.jdbc.Driver
 url :: jdbc:mysql://</logicalDB name> (for Local MySQL DB s/w)
 jdbc:mysql://<hostname>:<port no>/<logicalDB name> (for remote MySQL DB s/w)

host name or ipaddress machine
 and, port no of MySQL DB s/w where
 it is running

jar file :: mysql-connector-java-8.0.21.jar (supports auto-loading of driver class name)

(collect from internet --> mavenrepository.com)
<https://mvnrepository.com/artifact/mysql/mysql-connector-jar/8.0.21>

Procedure to make Java App interacting MySQL DB s/w

step1] keep any old JDBC App ready...

step2] add Connector/J JDBC driver jar file to build path/classpath..

step3] Modify the following details in the Application...

(a) driver class name
 (b) JDBC URL (c) DB Username (d) DB Password

//register JDBC driver s/w

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

/establish the connection

Connection con=DriverManager.getConnection("jdbc:mysql:////ntaj414db","root","root");

(or)

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ntaj414db","root","root");

When DB s/w is changed for JDBC App the following

JDBC properties will change

a) driver class name (optional)

b) JDBC URL

c) DB Username d) DB Pwd e) JDBC driver jar file..

note:: The class names of JDBC objects will change based on the JDBC Driver s/w and DB s/w we use ,So we never refer them with real class names related reference variables.. we always refer them with JDBC API supplied common interface reference variables .. to achieve loose coupling or run time polymorphism..

Candidate key column , natural key column ,surrogate key column and primary key column

Candidate key column

Db table in oracle

```
|-->pid (n) [ck]
|-->pname
|-->mobileNo [ck]
|-->aadharNo [ck]
|-->voterId [ck]
|-->panNo [ck]
|-->address
|-->bankAcNo [ck]
|-->passPortNo [ck]
|-->age
```

Candidate key column (ck)

The column in Db table ,whose values can be used to identify and access the records is called candidate key column.. This column contains unique value.

Natural key column

=>The candidate key column whose values are having business meaning and will be changed becoz of the outside world business policies are called natural key columns..
=>These column values are expected from endusers.
=>These column values can be used inside and outside and DB s/w to identify and access the information..
e.g: aadharNo, voterId ,panNo, bankAccount ,mobileNo and etc..

Surrogate key column

=>The candidate key column whose values are dynamically generated by the underlying Db s/w or App or Project. with out having any business meaning is called Surrogate key column..
=>these column values will not be from endusers... will be generated dynamically by underlying Db s/w or Project/App:
eg: => sequence / generated number in oracle
=>identity column [auto increment] number in mysql
=>hibernate generators
=> random number given by App/project

Limitations of taking natural key column as Primary key column

=> values are quite lengthy , so they need more memory
=> These values are having business meaning and they will be changed becoz of outside business policies and this process distrubts other dependent db tables and java classes (Very costly)
=>These values are expected from enduser.. If enduser fails to gives record insertion will fail..

note: Do not take natural key col as primary key column .. always prefer taking surrogate key column as pk column.

Advantages of taking surrogate key column as pk column

=> Values are very small, So much memory is not required
=> No business meaning.. so they never change
=>not expected from endusers.. record insertion is still possible through enduser is not supplying certain Data..

App development using Oracle sequence generated as the sno column value (pk col value)

step1) create sequence in oracle

```
SQL> create sequence stud_sno_seq start with 1000 increment by 1;
Sequence created.
```

=> natural key column, surrogate key column candidate key columns are theoretical constraints.. b/c they can not be applied
DB tables ...
=>The only possible real/practical constraints are pk, fk, uk, rnk ..

step2) develop insert query based application using PreparedStatement obj. having insert SQL query sQL query as shown below..

```
private static final String INSERT_STUDENT_QUERY="INSERT INTO STUDENT VALUES(STUD_SNO_SEQ.NEXTVAL,?,?);"
```

1.2.5

note:: DB sequence is not aware of wheather pk column values given by sequence is inserted or not , deleted or not ... So it always generates next number... in sequence..

note:: Oracle supports sequences .. but mysql does not support sequences, So mysql we use Identity col as pk value..

[The col having auto increment constraint , If insert values to other cols the identity col value will be generated automatically.]

In MySql Db s/w to generate sno values by talking "sno" col as identity col (standard)

step1) make sno column as identity column by add auto increment constraint
mysql workbench --> go student db table settings --> go to sno col and
select AI [auto increment] -->apply..

step2) develop application using PreparedStatement obj having following sql Query..

```
private static final String INSERT_STUD_QUERY="INSERT INTO STUDENT(SNAME,SADD,AVG) VALUES(?, ?, ?);"
```

note:: the identity col of mysql first uses <max val+1 formula to generate the id value.. and next uses previous val+1 formulae

To generate sno col value as random number through java code (for any Db s/w) (not standard)

```
private static final String INSERT_STUD_QUERY="INSERT INTO STUDENT VALUES(?, ?, ?);"
```

```
if(con!=null)
    ps=con.prepareStatement(INSERT_STUD_QUERY);
    //set values to query parametr?
    if(ps!=null) {
        ps.setInt(1, new Random().nextInt(1000000); // 0 to 99,999
        ps.setString(2,name);
        ps.setString(3,addr);
        ps.setFloat(4,avg);
    }
```

Random class generated random number
are not unique numbers.. For unique number
develop a custom code or use third api..

Interacting with multiple DB s/w's

==> One Java App can interact with multiple DB s/w's as needed...

=> In one Jdbc App we can create multiple JDBC con objects, Statement objects, PreparedStatement objects, ResultSet objects and etc...

=> In Jdbc App use Simple Statement object to execute SQL query only for 1 time, without having any conditions... PreparedStatement is good multiple operations like executing query for one time or multiple times with conditions... for multiple times with out conditions.

for date values insertion... for bid insertion and etc...

=> Usecase: Bank 1 has some records of bank1 will be transferred to another bank2

e.g.: Andhar Bank has acquired 100% vypne bank

e.g.: Andhar is merged with Union bank

e.g.: Syndicate bank is merged with Canara bank.

Q) Write JDBC App to transfer the records of db table [oracle] to db table [mysql]

(Just assume syndicate is acquired by school2)

setup

=====> BankAccount db table in oracle (AndharBank)

=====> BankAccount db table in mysql with records (Union bank)

BankAccount db table oracle

sn(pk)[1]	accno(pk)[2]	rn([1])	holdername[vc1]	balance[float]
1	34561	raja	7000	7000
2	56778	rani	8000	8000

BankAccount db table mysql

sn(pk)[1]	accno(pk)[2]	rn([1])	holdername[vc1]	balance[float]
1	12345	raresh	69000	69000
2	34567	raresh	70000	70000
3	34561	raje	7000	7000
4	56778	rani	8000	8000

as->not null
pk ->primary key AI ->Auto increment
rk ->unique key
One db table can have only one pk column

select * from BankAccount;

(Query with ext condition for 1 time execution)

Java App

read from [Simple Statement obj]

we need to execute

Insert into BankAccount(accno,holdername,balance) values(?, ?, ?)" for multiple times with diff Inputs. So use PreparedStatement obj.

io/bank account numbers are unique atleast within a country..

=> To get records from oracle DB table we write "Select * from BankAccount" SQL query.

since this SQL query is not having any conditions and needs to be executed only for 1 time. So we can use SimpleStatement object. To insert these record bulk records into mysql DB table we need to execute Insert SQL query for multiple times with diff values. For this we need to use PreparedStatement object.

Example App

refer to OracleToMySQLDataTransferTest.java

with Working date values

=====> while working DBs, DDL,DML,DDLC, JdbcStatement, registration date and etc., values we need to insert and receive date values.

=> Inserting date values as String col values [varchar2] is bad practice bcz we can not perform Arithmetic and logical operations on them

=> Insert date values to the "date" data type db table calls.

=> All major DB s/w's are supporting "date" data type...

=> Diff DB s/w's stores date values in diff patterns

oracle -> dd-MMM-yyyy eg: 20-OCT-88
mysql -> yyyy-MM-dd eg: 1990-08-15

note: Using Simple Statement we can insert date values but the SQL query should have date values directly in the pattern that underlying DB s/w supports, which very complex to supply, bcz diff ent user belong to diff countries will date values in diff patterns

India -> dd-MM-yyyy eg: 30-12-1988 MM -> month
usa -> MM-dd-yyyy eg: 12-10-1988 mm -> minutes..
china -> yyyy-MM-dd eg: 1980-12-31

IO Stream (read)

and etc..

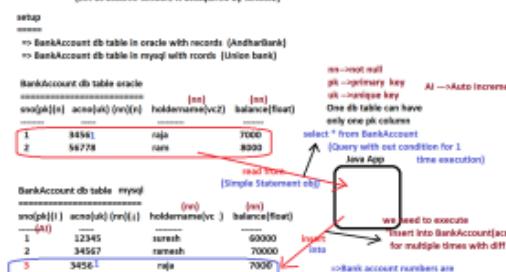
Interacting with multiple DB's w/o

=> One Jdbc App can interact with multiple DB's/w/o needed...

- > In one Jdbc App we can create multiple Jdbc conn objects, Statement objects, PreparedStatement objects, ResultSet objects and etc...
- > In Jdbc App use Simple Statement object to execute SQL query only for 1 time with out having any conditions... PreparedStatement is good multiple operations like executing query for one time or multiple times with conditions ... for multiple times with out conditions.
- > for date values insertion , for bid insertion and etc...

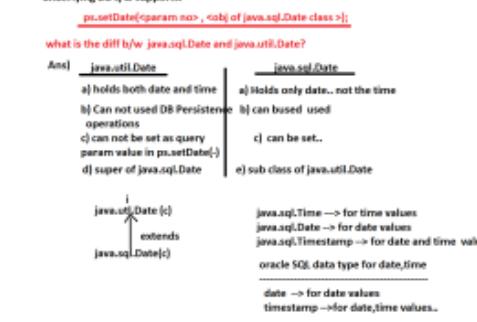
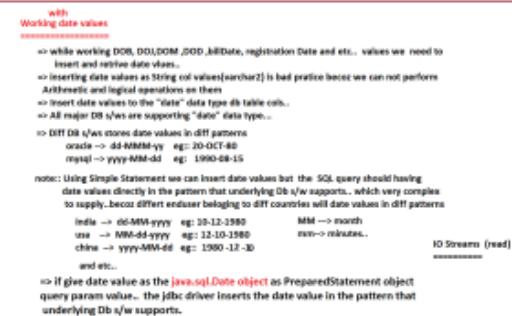
=>UseCase: Bank1 has acquired Bank2 then the records of bank1 will be transferred to another bank2
e.g:
1) Ashra is merged with Union bank
2) Syndicate bank is merged with city bank

Q) Write JDBC App to transfer the records of db table [oracle] to db table [mysql]
(Let us assume school1 is acquired by school2)



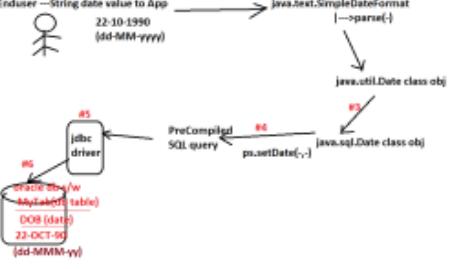
> To get records from oracle DB table we write "Select * from BankAccount" SQL query .. since this SQL query is not having any conditions and needs to be executed only for 1 time, so we can for SimpleStatement object.. To insert these received bulk records into mysql DB table we need to execute insert SQL query for multiple times with diff values .. for this we need to use PreparedStatement object..

Example App
refer to OracleToMySQLDataTransferTest.java



Standard Procedure: jdbc driver independent and DB s/w independent

in JDBC App to development to insert date values to db table cols



```

//converting String date value to java.util.Date class obj
String d1="12-31-1990"; //dd-MM-yyyy
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy"); //holding pattern of date
java.util.Date ud1=sdf.parse(d1);
System.out.println(ud1);

=>always write java.util.Date and
java.sql.Date classes with package names
in the .java file to avoid ambiguity errors.

```

```

//converting java.util.Date class obj to java.sql.Date class obj
long ms=ud1.getTime(); //gives no of milliseconds that are elapsed b/w
                     //jan 1st 1970 midnight 00:00 hours(epoch standard) and ud1 obj's date, time
                     //java.sql.Date obj uses java.util.Date(ms);
System.out.println("sql date"+ud1);
In s/w industry, all date and time
will be remembered as millis with respect
to the epoch standard 1970 jan 1st midnight 00:00 hours

```

If String date value pattern is `yyyy-MM-dd` then it can be converted directly to
`// java.sql.Date class obj with out converting to java.util.Date class object.`
`String d2="1995-10-21"; //yyyy-MM-dd`
`java.sql.Date sd2=java.sql.Date.valueOf(d2); //static method`
`System.out.println("sql date"+sd2);`

Write a JDBC App to insert DOB,DOM ,DOJ date values from user along with name, and address



CREATE TABLE PERSON_DATE_TAB1;

| PID NUMBER(10) PRIMARY KEY,
PNAME VARCHAR2(20),
PADDRS VARCHAR2(20),
DOB DATE,
DOI DATE,
DOM DATE;|

CREATE SEQUENCE PERSON_PID_SEQ

MINVALUE 1 MAXVALUE 1000000 INCREMENT
BY 1 START WITH 1

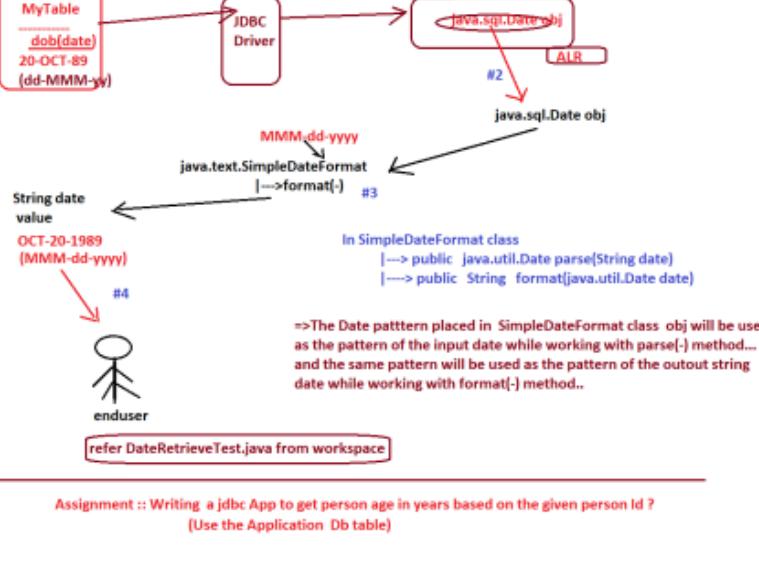
Refer DateInsertTest.java from workspace

Retrieving Date values

=>For this we can use either SimpleStatement object or PreparedStatement obj
=>if u r just reading and display date values.. with out taking date value as the condition value
the prefer Simple Statement object/PreparedStatement object, otherwise prefer using
PreparedStatement obj.
=>After getting Date values to ResultSet object by executing Select SQL query ,if we call
rs.getString(-) the date values will come as string values having yyyy-MM-dd pattern .If
we call rs.getDate(-) method then we get java.sql.Date class objs having yyyy-MM-dd pattern
as internal pattern.
=>calling rs.getDate(-) method is good becoz we can use the received java.sql.Date class obj
to present date in different formats as required for the enduser .. by using
format(-) method of SimpleDateFormat class.



DB s/w independent and JDBC driver independent procedure to retrieve and display
date values from date data type DB table cols



Assignment :: Writing a jdbc App to get person age in years based on the given person Id ?
(Use the Application Db table)

[refer DateRetrieveTest.java from workspace]

To calculate age based on given person id

- =====
- a) Using DB s/w specific SQL query
 - b) Using Java code {DB s/w independent code}

a) Using DB s/w specific SQL query

query in oracle (refer PersonAgeCalculatorOracle.java)
=====

SQL> select (sysdate-dob)/365.25 from person_date_tab1 where pid=1;

(SYSDATE-DOB)/365.25

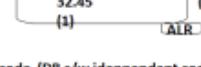
20.7739647

query in mysql (refer PersonAgeCalculatorMySQL.java)

mysql> select from_days(datediff(now(),dob)) from person_date_tab1 where pid=1;

[or]

select datediff(now(),dob)/365.25 from person_date_tab1 where pid=1;



b) Using Java code {DB s/w independent code}

Write java code for this

($\text{sys date in Ms} - \text{DOB date in Ms} / (1000 * 60 * 60 * 24 * 365.25)$)

(refer PersonAgeCalculatorJavaCode.java)

```
if(rs!=null){  
    if(rs.next()) {  
        // get System date in ms  
        sysDateMs=new Date().getTime();  
        // get DOB from from ResultSEt  
        udob=rs.getDate(1); // java.util.Date class ref variable (udob) is referring java.sql.Date class obj  
        //as super class ref can refer sub class obj  
        dobMs=udob.getTime();  
        System.out.println("Person age::"+{sysDateMs-dobMs}/{1000*60*60*24*365.25f});  
    }  
    else {  
        System.out.println("Record not found");  
    }  
}//if
```

Working with Large Objects (files)

Large objects(LOBs/Files)

BLOB
(Binary Large object)
(eg: image files, audio files, video files, av files, and etc...)

CLOB
(Character Large Object)
(eg: text files, Rich Text files (RTF), csv file and etc..)

av files :: audio video files
csv file :: coma seperated values

=> All major DB s/w are giving support for LOBs by supporting data types called BLOB and CLOB

=> while working with matrimonial apps(match makings), job portal Apps (like naukri.com), social networking Apps (like fb), profile management Apps (aadhar Apps), e-commerce apps and etc.. we need to insert and retrieve large objects...

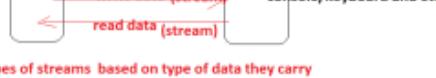
=> Simple Statement object SQL query expects files(lobs) directly as input values.. but that not possible.. So we can not use Simple Statement obj to insert large objects

=> PreparedStatement object allows to set streams representing files as query param values .. So we can use PreparedStatement object to insert Large objects...

note:: The SimpleStatement obj SQL query does not even allow streams as inputs values.. it expects total file content as the input value.. which is again not possible..

Streams

=> Stream is a continuous flow of data.. and acts as pipelines b/w source and destination using we can send/receive to/from Destination.



Two types of streams based on type of data they carry

[a] Byte Streams

=> These streams can carry both binary and text data ..

=> Here InputStream are given for reading data and OutputStream given for writing data

eg:: FileInputStream , FileOutputStream, DataInputStream, DataOutputStream and etc...

[b] Character streams

=> These streams can carry only text data ..

=> Here Reader Streams are given for reading data and Writer Streams given for writing data

eg:: FileWriter , FileReader, BufferedReader, BufferedWriter and etc...

binary

To set input streams representing files as query param value we can use

ps.setBinaryStream[-,-] / ps.setBinaryStream[-,-] for binary data..

ne.setRinhl[-,-]/ne.setRinhl[-,-] (or) 1,2 1-->param index

1,2,3

2-->stream 3-->length of file (from the beginning)

text

To set Reader streams representing files as query param value we can use

ps.setCharacterStream[-,-] / ps.setCharacterStream[-,-] (or) for character/text data..

ps.setClob[-,-] / ps.setClob[-,-]

1,2,3

1,2

JDBC driver and DB s/w independent Procedure to insert image (BLOB value) to DB table col

step1] Create PreparedStatement having Pre-compiled SQL query

```
PreparedStatement ps=
  con.prepareStatement("INSERT INTO SHADHI_INFO VALUES( PID_SEQ.NEXTVAL, ?, ?, ? )");
```

step2] Create ByteStream representing image file of the file system

InputStream is=new FileInputStream("d:/images/photo.jpg");

step3] set values to query params

```
ps.setString(1,"ramesh");
ps.setString(2,"hyd"); ps.setFloat(3,600000);
ps.setBinaryStream(4,is);
```

step4] execute the query

int count=ps.executeUpdate();

photo.jpg (d:/images/photo.jpg)



(Since lobs are stored directly in Db s/w ...
the burden on Db s/w will increase... So performance is bad)

=> In oracle Db s/w each blob /clob value can be upto 4gb

=> mysql Db s/w can not huge size blob or clob values...

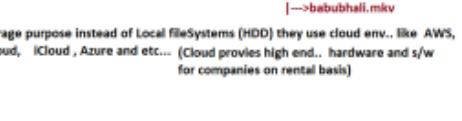
Matrimonial App (standalone)-->Go for BLOB,CLOB values
(Jdbc Apps) [super market]
(college Apps, small university Apps)

Matrimonial App (web based)-->Go for file system/cloud storage
(Servlet/Jsp env...) for files and insert only locations

(youtube, FB, Instagram, twitter and etc.. will follow this model)

more eg:amazon,flipkart ..

note:: Storing LOBs (files) directly in DB s/w will be done only small scale Apps.. not for large scale Apps ... In large scale Apps files will be saved in file system (HDD) and their locations will be saved in DB table cols as String values.



=> For storage purpose instead of Local fileSystems (HDD) they use cloud env.. like AWS, GoogleCloud, iCloud, Azure and etc... (Cloud provides high end.. hardware and s/w for companies on rental basis)

=> \"\\" is escape sequence character in java ... it will be useful to void literal meaning certain for characters like " , " and etc.. to make them as printable characters..

\\"-->gives "
\'--> gives '
\--> gives \
and etc..

mysql blob data type are

TINYBLOB Up to 255 bytes
BLOB Up to 64 Kb
MEDIUMBLOB Up to 16 Mb
LONGBLOB Up to 4 Gb

oracle having only blob data type having size upto 4gb

JDBC driver and DB s/w indepednet procedure to retrieve BLOB values to DB tables

Here we can use either simple statement or
PreparedStatement object

oracle Db s/w shadhi_tab			
regNo	name	location	income
1	raja	hyd	900000



step1] create PreparedStatement obj having pre-compiled SQL query

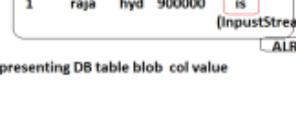
PreparedStatement ps= con.prepareStatement("SELECT * FROM SHADHI_TAB WHERE REGNO=?");

step2] set Query param values..

ps.setInt(1,1001);

step3] execute the query

ResultSet rs=ps.executeQuery();



step4] get InputStream from ResultSet representing DB table blob col value

InputStream is=null;

If(rs.next())

is=rs.getBinaryStream(5);

step5] create OutputStream pointing new file

OutputStream os=new FileOutputStream("new_pict.jpg");



step6] copy data of InputStream to OutputStream file

IOUtils.copy(is,os);

(It is from third party api given apache company

jar file name:: commons-io-<ver>.jar)



(Download from internet)

(url :https://mvnrepository.com/artifact/commons-io/commons-io/2.7)

=>IOUtils is third party supplied Utility class to simplify IO streams related operations..

=>The class that acts as helper class while doing major operations.. convert 1 form of data to another form data.. copying data and etc.. is Utility class

eg:: IOUtils , DateUtils, StringUtils, BeanUtils, Arrays and etc....

step7] close jdbc objs and streams

is.close(); os.close(); rs.close(); ps.close();

con.close();

note:: refere PSBLORetrieve.java

=>To insert blob value to DB table column we need set Reader Stream that pointing to text file as

Query param value by using `ps.setCharacterStream(-,-)`/`ps.setClob(-,-)` or
`ps.setClob(-,-)/ps.setClob(-,-)` methods

=>For retrieving blob values from DB table col through JDBC ResultSet object us

`rs.getCharacterStream(-)` that given Reader object..

=>To insert pdfs files,word docs prefer using BLOB cols becoz those file contain both text and images

=> To Insert text files, csv [ms excel], rich text files prefer using CLOB cols ..

Mysql CLOB data types are

=====

TINYTEXT upto 256 bytes

TEXT upto 65,535 bytes

MEDIUMTEXT upto 16,777,215 bytes

LONGTEXT upto 4,294,967,295 bytes

Assignment :: develop Profile Mgmt Application to insert PHOTO,RESUME,VIDEO in a single DB table column..

PostgreSQL

=====
=>type :: DB s/w [Light weight] default username:: postgres

=>vendor:: EnterpriseDB password :: root (choose during installation)

=>default portno:: 5432

=>Allows to create logical DB , default logical name:: postgres

=>Gives PgAdmin4/5 as built-in GUI DB tool

=>To download and install :: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Procedure to create DB table with records in PostgreSQL using PGadmin tool

=====

step1) make sure that PostgreSQL Db s/w is installed.

step2) launch pgadmin4/5 tool (use start button ---> all prgs ---> postgresql ---> pgadmin4/5 ---> submit password ->...)

step3) create Logical DB

right click on databases ---> create database ---> name :: ntaj414db --->

step4) create DB table product

expand ntaj414db---> expand schemas --->expand --->public --->right click tables--->new
table name :: product --> go to columns tab -> add cols
pid inteteger pk {yes} nn{yes}
pname charcter varying 20
price real
qty real
---> ok....

generated SQL is

```
CREATE TABLE public.product
{
    pid integer NOT NULL,
    pname character varying(20) COLLATE pg_catalog."default",
    price real,
    qty real,
    CONSTRAINT product_pkey PRIMARY KEY (pid)
}
```

=> To insert blob value to DB table column we need set Reader Stream that pointing to text file as Query param value by using `ps.setCharacterStream(-,-)` or `ps.setClob(-,-)` methods

=> For retrieving blob values from DB table col through JDBC ResultSet object us `rs.getCharacterStream(-)` that given Reader object..

=> To insert pdfs files,word docs prefer using BLOB cols becoz those file contain both text and images

=> To Insert text files, csv [ms excel], rich text files prefer using CLOB cols ..

Mysql CLOB data types are

TINYTEXT upto 256 bytes	refer CLOBInsertTestOracle.java,CLOBInsertTestMySQL.java and
TEXT upto 65,535 bytes	CLOBRetrieveTest.java
MEDIUMTEXT upto 16,777,215 bytes	
LONGTEXT upto 4,294,967,295 bytes	

Assignment :: develop Profile Mgmt Application to insert PHOTO,RESUME,VIDEO in a single DB table column..

PostgreSQL

=> type :: DB s/w [Light weight] default username:: postgres
=> vendor:: EnterpriseDB password :: root (choose during installation)
=> default portno:: 5432

=> Allows to create logical DB , default logical name:: postgres

=> Gives PgAdmin4/5 as built-in GUI DB tool

=> To download and install :: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Procedure to create DB table with records in PostgreSQL using PGadmin tool

=====

step1) make sure that PostgreSQL Db s/w is installed.

step2) launch pgadmin4/5 tool (use start button --> all prgs --> postgresql --> pgadmin4/5 --> submit password ->...)

step3) create Logical DB

right click on databases --> create database --> name :: ntaj414db --> ...

step4) create DB table product

expand ntaj414db--> expand schemas -->expand -->public -->right click tables-->new

table name :: product --> go to columns tab -> add cols

pid integer pk {yes} nn{yes}	pname charcter varying 20
price real	qty real

--> ok...

generated SQL is

```
CREATE TABLE public.product
{
    pid integer NOT NULL,
    pname character varying[20] COLLATE pg_catalog."default",
    price real,
    qty real,
    CONSTRAINT product_pkey PRIMARY KEY (pid)
}
```

step5) insert records into db table...

right click on product db table -->scripts --> insert query --> replace values for ? symbols

--> execute query (flash symbol - 

PostgreSQL supplied type4 mechanism based jdbc driver s/w for postgresql Db s/w is called postgresql thin driver and its details are

driver class name :: org.postgresql.Driver

jdbc url :: jdbc:postgresql:<logical DB> {for Local postgresql Db s/w}

jdbc:postgresql://<host>/ip address>:<port>:<LogicalDB> {for remote postgresql db s/w}

jar file :: postgresql-42.2.16.jar (supports auto loading driver class) seperate download from internet

 mvnrepository.com

(<https://mvnrepository.com/artifact/org.postgresql/postgresql/42.2.16>)

Procedure to develop Java App interacting with PostGreSQL DB s/w

=====

(a) keep any old app ready and make sure that its DB table is available in postgresql.

b) Write the following to create jdbc con object pointing to PostgreSQL Db s/w..

```
// register jdbc driver
//Class.forName("org.postgresql.Driver");
//establish the connection
Connection con=DriverManager.getConnection("jdbc:postgresql:ntaj414db","postgres","root");
          jdb url           dbuser      dbpwd
```

c) Change SQL query as per PostgreSQL DB s/w..

```
ResultSet rs=st.executeQuery("SELECT PID,PNAME,PRICE,QTY FROM PRODUCT");
```

d) add postgresql thin jdbc driver s/w jar file to classpath or build path...

right clikc on the project-->build path-->cfg buildpath -->libraries tab -->add extenal archvies --> browser and select the jar file.. (postgresql-42.2.16.jar [select from downloaded files location])

e) run the Application

note:: In Postgresql's pgadmin take all names like db table name, col names, sequence names and etc.. in lowercase.. but in front end App we can them in the SQL queries in any case.. (generally we prefer in upper case)

note:: Postgre sql supports sequences i.e we sequence generated value as pk col value while inserting record to db table cols.. and the qury sequence looks like this..

```
private static final String INSERT_QUERY="INSERT INTO PRODUCT VALUES(nextval['PID_SEQ'],?, ?, ?);"
```

 (to sequence generated value)

To create sequence in PostgreSQL

=====

```
u r logical DB -->schemas -->sequences-->create sequence -> name: pid_seq -->definition tab -->
current value :: 1000 , min value : 1 , increment :1 end value : 1000000 -->.... -->execute [flash symbol]
//Generated Qury
CREATE SEQUENCE public."PID_SEQ"
INCREMENT 1
START 1000
MINVALUE 1
MAXVALUE 100000;
```

Assignment:: Develop Apps by choosing PostgreSQL as Db s/w

(a) insert Date values (like DOB,DOM,DOJ)

(b) insert blob, clob values

(c) calculating age based given person id and his DOB

Also retrive operation

=====

prepare and come for next class

=====

=> oracle/mysql PL/SQL procedures and functions

=> oracle Cursors (Sys_refcursor)

CallableStatement object [It is given to call PL/SQL procedure and functions]

PL/SQL Procedures/Functions

=====

=>These are named blocks that reside and execute in DB s/w havig block of SQL and Non SQL Statements .. These are like java methods..

=> PL/SQL procedure does not return a value.. where as PL/SQL function returns a value

=> PL/SQL (PL by SQL) Procedure or function params will have mode and type

3 modes are

IN --> input (To supply inputs)

OUT --> To gather outputs

INOUT --> To supply inputs and to gather outputs

pl/sql logic in oracle

in SQL or PL/SQL

= for comparision

:= for assignment

y:=x+x; x --> in param , y out param

pl/sql logic in oracle

x:=x*x; x--> INOUT param

note:: if want 10 results from PL/SQL procedure then we need to take 10 OUT params.. To do same thing with PL/SQL function we need to use 9 OUT params and 1 return value.

=>PL/SQL programing syntax is specific to each DB s/w...

=> These are also called as Stored Procedures and functions becoz they reside and execute in DB s/w..

=> Project

|--->modules

|--->Apps

|--->prgs..

=>Instead of writing same SQL query in multiple modules or Apps , It is recommended to write in DB s/w only for 1 time as PL/SQL procedure or function and use it from multiple apps or modules having reusability.. It helps to centralize PErsistence logic by keeping in Db s/w with protection (i.e we can hide code from developers)

usecase1: Instead authentication logic, attendant calculation logic in every module , Write it only for 1 time in Db s/w as PL/SQL Procedure or function and use in multiple modules..

In realtime project

=====

=> 70% to 80% PErsistence logic in the form of SQL queries

=> 20% to 30% PErsistence logic in the form of PL/SQL procedure and functions

↓
Simple PL/SQL procedure/functions
should be developed by Programmers directly..
where complex procedures and
functions will be developed by SQL/DB Team.

=> JDBC Apps use CallableStatement obj to call PL/SQL procedures and functions..

↓
This can be even used to execute static /dynamic SQL queries
and pre-compiled SQL queries

java Project contains

|--->UI developers (html,css,js,angular,...)
|--->Java Developers (java, servlet,jsp, spring,hib,...)
|--->SQL developers (SQL, PL/SQL, DBA ...)

Sample /Example PL/SQL procedure in oracle using SQL Developer

right click on procedure ---> create procedure ---> name:: P_FIRST ---> add{+} 3 params

x in number ,y in number ,z out number

--> save (flopy disk) ---> compile (settings symbol -gear)

CREATE OR REPLACE PROCEDURE P_FIRST(x IN NUMBER , y IN NUMBER , z OUT NUMBER) AS

BEGIN

z:=x+y;

END ;

procedure
name
param
name
param
mode

default mode of param is "IN"

{or}

Process to create PL/SQL Procedure by using SQL command propmt / SQL Plus

=>open SQL prompt as administrator --->SQL> connect system/manager --->SQL> select * from emp; (some summary)
---> ed ---> type the the above PL/SQL procedure ---> file menu save ---> file menu exit ---> SQL>/

if any compilation error are there ---> SQL> show errors;

Steps to use Callable Statement obj to call PL/SQL procedure

step1] prepare Query having SQL92 syntax to call PL/SQL procedure

String query="(CALL P_FIRST[?, ?, ?])";

1,2,3

step2] Create CallableStatement obj by making the above Query as Pre-compiled SQL query

CallableStatement cs=con.prepareCall(query);

sends the given SQL to DB s/w , makes it as PL/SQL
procedure or function and returns CallableStatement obj
having the pre-compiled SQL query.

java.sql.Statement()

↑ extends

java.sql.PreparedStatement()

↑ extends

java.sql.CallableStatement()

step3] register OUT params with JDBC data types.

cs.registerOutParameter(3, Types.INTEGER);

param index

jdbc data type

JDBC data types are bridge data types b/w java data types
and SQL Data types they are useful to give instructions to JDBC driver to
convert java notation data to SQL notation data and vice-versa,

step4] set values to IN params

cs.setInt(1,100); no need of registering IN params
cs.setInt(2,200); with JDBC type.. these cs.setXxx(-,-)
will register them internally..

java datatype

jdbc type

SQL data type (oracle)

step5] call /execute PL/SQL procedure

int

float

short

java.sql.Date

String

Types.INTEGER

Types.FLOAT

Types.SHORT

Types.DATE

Types.VARCHAR

number

float , number(-,-)

number

date

varchar,varchar2

cs.execute(); [this uses the above SQL query with special
syntax to call PL/SQL procedure.. In the process
output will be stored in OUT params]

and etc..

All these are constants
(public static final variables)
of java.sql.Types class.

step6] get results from OUT Params

int result=cs.getInt(3);
System.out.println("result is::"+result);

note:: To execute PL/SQL procedure for multiple times with diff values repeat step4 to step6 for multiple times

step7] close jdbc objs
cs.close(); con.close();

Write a JDBC App that calls PL/SQL procedure of Oracle using CallableStatement obj to get Emp name,job,salary based on the given number..

```
create or replace procedure p_getEmpDetails_By_Id(  
    name_in varchar,  
    job_in varchar,  
    salary_out number);  
  
as  
begin  
    select ename,job,sal into name,job,sal from emp where empno=:id;  
end;  
/
```

For Java App refer C5ProcedureTest2.java

Q) Write a JDBC App to call PL/SQL procedure using CallableStatement object to perform Authentication...

step1] make sure userinfo db table is already available having usernames ,passwords.

SQL> select * from userinfo;

UNAME	PWD
raja	raji
ramesh	ramesh
jani	begum
shani	black

step2] develop PL/SQL procedure having Authentication logic..

```
Create or replace procedure P_AUTHENTICATION(user_in varchar,  
pass_in varchar2,  
result_out varchar2)  
as  
cnt number;  
begin  
select count(*) into cnt from userinfo where uname=user and pwd=pass;  
if(cnt>0)then  
    result:=VALID CREDENTIALS;  
else  
    result:=INVALID CREDENTIALS;  
end if;  
end;  
/
```

step3] develop the App using CallableStatement object

refer C5ProcedureTest3.java

==Since CallableStatement object is the extension of PreparedStatement obj.. So, there is not SQL injection Problem with CallableStatement object..

Assignment1: get StudentDetails based on the given sno by Calling PL/SQL procedure.

Assignment2: get Dept Details based on the given deptno by calling PL/SQL procedure.

Cursor

=> It is a InMemory variable of oracle's PL/SQL programming having ability to store batch of records given by Select SQL query. Cursor is like java ResultSet object.
=> SYS_REFCURSOR is a built-in Cursor data type.. In oracle's PL/SQL programming

```
empDetails sys_refcursor; //cursor variable declaration
open empDetails for
  select empno,ename,job,sal from emp; //The records given by
                                         //select SQL query execution will
                                         //be stored in cursor variable "empDetails"
```

Write a PL/SQL procedure that gives Employee(empno,ename,job,sal) details based on given given two design, order by design.

```
CREATE OR REPLACE PROCEDURE P_GET_EMPDETAILS_BY_DESIGN
(
  DESG1 IN VARCHAR2 | (in param)
 ,DESG2 IN VARCHAR2 | (in param)
 ,EMPDETAILS OUT SYS_REFCURSOR | Cursor type out param
 )AS
BEGIN
```

the records given by select query [] will be stored here

END ;

In java.sql.Types class we do not have JDBC data type to register Cursor type Out params.
So we need to use OracleTypes.CURSOR as alternate [given inojdbc7/Types/S.java file]

```
cs.registerOutParameter(1,OracleTypes.CURSOR);
--so get refcted from sys_refcursor type out parameter by calling
c.getObject() which actually gives JDBC Resultset object...
```

ResultSet rs=ResultSets.getObject(1);

For example app refer CProcedureCursorTest.java

prepare try with resource feature of jdk?

Calling PL/SQL function by using CallableStatement object

=> PL/SQL function return a value.
=> If PL/SQL function wants to return 10 results.. then it gives 9 results through OUT params
one result through return value.
=> SQL 92 syn :- { ?call character-name(?,?,...)}

1 2,3,4,...

=> return param (1->?) should be treated as out param and must be registered JDBC data type..

Write a Oracle PL/SQL function to get Student name,addr,avg based on the given Student number

Q

3 results [In PL/SQL function 2 out params , 1 return value]

```
CREATE OR REPLACE FUNCTION FX_GET_STUD_DETAILS_BY_SNO
(
  NO IN NUMBER
 ,NAME OUT VARCHAR2
 ,ADDR OUT VARCHAR2
 ,AVERAGE OUT NUMBER
 )RETURN ELOB AS return type
  PERFORMANCE FLOAT;
  LOCAL VARIABLE
BEGIN
  SELECT SNAME,SADD,AVG INTO NAME,ADDR,PERFORMANCE FROM STUDENT WHERE SNO=NO;
  call names          out param   local variable name    db table      in param
  return PERFORMANCE;          local variable       name
END;           return statement returning the value..
```

SQL 92 syntax is :: [?] call FX_GET_STUD_DETAILS_BY_SNO(?,?)

1 2,3,4

refer CFunctionTest5.java

```
[?] call FX_GET_STUD_DETAILS_BY_SNO(?,?)
```

1 2,3,4

Assignment1:: write a JDBC App that calls PL/SQL function to get Emp details(name,job,sal/design) by given number.. [java]

Assignment2:: write a JDBC App to get Student details by calling PL/SQL procedure based on given initial chars of student name.. [cursor]

Calling PL/SQL procedures and functions

In oracle

>>SYS_REFCURSOR ----> built-in cursor data type (To hold bunch of records given by select Query)
>>SQL%ROWCOUNT --> returns the numeric value representing the nof records that are effected when non-select SQL query is executed.

note: we can have both select and non-select SQL queries in a PL/SQL procedure.

example PL/SQL function in oracle to View and delete student records:

```
CREATE OR REPLACE FUNCTION FX_VIEW_DELETE_STUD_BY_SNO
(
  NO IN NUMBER
 , STUDODTLS OUT SYS_REFCURSOR
 ) RETURN VARCHAR2(30)  | return type
 CNT NUMBER;           | local variables
 RESULT VARCHAR2(30); | local variables

BEGIN
OPEN STUDODTLS FOR
  SELECT SNO,NAME,SADD,Avg FROM STUDENT WHERE SNO=<NO> | holds record
                                                                | given select query
DELETE FROM STUDENT WHERE SNO=<NO>| no.of records deleted by delete SQL query will be stored in to cnt through SQL%ROWCOUNT
cnt=<SQL%ROWCOUNT>| 
IF (CNT=1) THEN
  RESULT:=>"RECORD DELETED";
ELSE
  RESULT:=>"RECORD NOT FOUND FOR DELETION";
END IF;
RETURN RESULT; | return statement returning the value.
END;
```



refer _FunctionTest6.java

Creating PL/SQL procedure in MySQL Using Workbench: to get prod_name,price,qty based on given pid

Launch workbench ---> expand mta444db ---> right click stored procedures --->create new procedure
DELMITTER \$\$ //start of procedure code
USE `mta444db`\$\$

```
CREATE PROCEDURE `P_GET_PROD_DETAILS_BY_PID` (in id int,out name varchar(30),
                                              out rate float,out qt float)
BEGIN
  select pname,price,qty into name,rate,qt from product where pid=<id> | mode >paramname< types
END$$ | col names   out param names    columnar IN param
```

DELMITTER ; //end of procedure code
Refer _CProcedureByID/Test7.java

note: No Cursors are given in mysql ... but no problem... we can bunch of records directly... with out having any support.

PL/SQL procedure in mysql that gives product details based given two product names

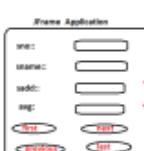
```
DELMITTER $$ | mode >paramname< types
USE `mta444db`$$
CREATE PROCEDURE `p_GET_PROD_BY_NAMES` (in name1 varchar(10),
                                         in name2 varchar(10))
BEGIN
  SELECT PID,PNAME,PRICE,QTY FROM PRODUCT WHERE PNAME IN(<name1>,<name2>);
END$$
```

DELMITTER ;
Query to call the above PL/SQL procedure :: [CALL p_GET_PROD_BY_NAMES(?,?)]

To get Select Query results from PL/SQL procedure using CallableStatement ob:
Resultset rs=ob.getResultSet();

In Java standalone first -> GUI front end -> Swing(JDK)

 In web application -> GUI FE -> [HTML/JS+CSS+JavaScript]



when the end user can click any button dynamically ... and we need to move the cursor in the RS dynamically. So it is recommended to take RS as scrollable RS.

- => Since there is no next record for last record, So call rs.last() method before calling rs.next()
- => Since there is no previous record for first record, So call rs.first() method before calling rs.previous()
- => Since the cursor can click any button dynamically ... and we need to move the cursor in the RS dynamically. So it is recommended to take RS as scrollable RS.

Container (Which holds the camp)

```

JFrame
=====
Components
-----
JTextField --> 4
JLabel --> 4
JButton --> 4
LayoutManager
-----
FlowLayout (keeps horizontally center)
Event
-----
ActionEvent (clicking on the button)
EventListener This provides env. to handle event (executing logic when event is raised)
----- by providing event handling method
ActionListener ( )
Event Handling method contains method deal
-----
```

`public void actionPerformed(ActionEvent ae)` --- this method will be called automatically when the ActionEvent is raised.

- => To develop Frame , take a class extending JFrame
- => To develop any Event Listener take a class implementing ActionListener() like ActionListener

```

public class ScrollFrame extends JFrame implements ActionListener{
  ...
  ...
  ...
}
```

[Refer_ScrollFrame_GUI.java](#)

<> In Java 10, the var keyword allows local variable type inference, which means the type for the local variable will be inferred by the compiler, so you don't need to declare it.

var Registration: [can be applied for local variable]

var dr = new Order();

dr.orderName = "apple";

dr.orderPrice = 100.75;

How change the Jdk version of eclipse project?

step[1] make sure new Jdk is/w/that u want to configure is installed in a computer

step[2] change to new jdk

right click on project --> build path --> library tab-->remove existing Jdk --> edit --> add -->

installed JREs --> add --> standard VM --> Directory (select Jdk new version installation folder)

-->



step[3] change compiler to new jdk version and also in project facets...

right click on project --> properties --> Java compiler :: 1.8



<> By adding GUI builder plugin to Eclipse IDE ... we can develop AWT/Swing app in drag and drop way...

highlight is a plugin which provides additional functionalities to existing JSE or App...

Eclipse is offering two types of plugins

a) Eclipse supplied plugins (can installable new JSE option of help menu)

b) 3rd party supplied plugins (can follow market place)

e.g. STS plugin , maven , git plugin and etc...

Procedure to install GUI builder plugin in eclipse 2018-12 IDE

Help menu --> install new software --> select 2018-12 - http://download.eclipse.org/releases/2018-12 -->

go to general purpose tool --> select all options that are having swing and SWT -- next --> next -->

accept terms and conditions --> restart ide ...

<https://www.youtube.com/watch?v=OULPbTYM2Sc&t=23s> ----- (installing GUI builder plugin...)

Procedure to develop scroll frame GUI App using Eclipse GUI builder

step[1] make sure that GUI builder plugin is installed refer this : <https://www.youtube.com/watch?v=QzokwHd1xjU>

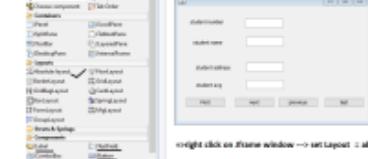
step[2] develop Class in project as JFrame / Application window class...

right click on package (newest file) --> new --> others --> search for JFrame/App window -->

give class name :: ScrollFrame_GUI_WFRE -- go to design mode..



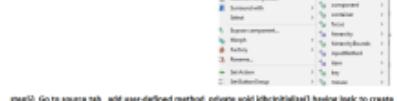
step[3] Design App by using drag and drop feature



step[4] Add action event handling to 4 button comps..

right click on button --> select-->

add event handling --> actionPerformed <-> ...



step[5] Go to source tab add user-defined method private void first() { } having logic to create

Statement R and call that method from actionPerformed and also declare necessary Jdbc related instance variables.

```
private Connection con;
private PreparedStatement ps;
private ResultSet rs;
private static final String GET_STUDENT_QUERY="SELECT * FROM STUDENT WHERE STUDENT_ID=?";
private static final String INSERT_STUDENT_QUERY="INSERT INTO STUDENT(STUDENT_ID,STUDENT_NAME,STUDENT_AGE,STUDENT_GENDER,STUDENT_ADDRESS,STUDENT_PHONE,STUDENT_EMAIL) VALUES(?,?,?,?,?,?,?,?)";
Connection con=null;
Statement st=null;
PreparedStatement ps=null;
ResultSet rs=null;
```

step[6] add "first" logic in actionPerformed() of first button. also for other buttons...

public void actionPerformed(ActionEvent e){

try{

rs.first();

textField1.setText(rs.getString(1));

textField2.setText(rs.getString(2));

textField3.setText(rs.getString(3));

textField4.setText(rs.getString(4));

}

catch(SQLException ex){

ex.printStackTrace();

}

step[7] run the Application...

- a) ScrollableRS
- b) InsensitiveRS
- c) UpdatableRS
- d) ReadOnlyRS

What is the difference b/w SensitiveRS and InsensitiveRS ?

Ans:- when RS object is representing DB table records.., if the modifications done is underlying DB table are reflected to RS immediately then it is called Sensitive RS... if not reflecting then it is called Insensitive.

Sensitive RS usecases :: getting stock market share values, displaying live game scores and etc..
Insensitive RS usecases :: report generation having static data like sales report , progress report and etc..

To create SensitiveRS

```
Statement stmt=con.createStatement();
Statement st=con.createStatement();
ResultSet rs=stmt.executeQuery("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
To create InsensitiveRS
Statement st1=con.createStatement();
ResultSet rs1=st1.executeQuery("SELECT * FROM STUDENT");
```

oracle thin driver,
mysql connector/j driver and etc..
these drivers supports but we need perform
few additional operations
=> we should avoid * from SELECT SQL Query
we should col names.
=> in each iteration of while[rs.next()] we should
place rs.refreshRow();

All jdbc drivers gives support
Insensitive RS with out any extra
settings.. by default Non-ScrollableRS
is Insensitive RS

Code for InsensitiveRS

```
Statement con.createStatement() ResultSet.TYPE_SCROLL_INSENSITIVE,
                           ResultSet.CONCUR_UPDATABLE);
//Create ResultSet object (scrollable)
if(rs!=null){
    rs.executeUpdate("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
    System.out.println(" Scrollable RS created...");

    if(rs.isHeldByMe()){
        while(rs.next()){
            if(count==0){
                Thread.sleep(30000); //during this period modify DB table records from SQL prompt/developer
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3));
                count++;
            }
        }
    }
}
```

Code for SensitiveRS

```
if(rs!=null){
    rs.con.createStatement() ResultSet.TYPE_SCROLL_SENSITIVE,
                           ResultSet.CONCUR_UPDATABLE);
//Create ResultSet object (scrollable)
if(rs!=null){
    rs.executeUpdate("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
    System.out.println(" Scrollable RS created...");

    if(rs.isHeldByMe()){
        while(rs.next()){
            if(count==0){
                Thread.sleep(30000); //during this period modify DB table records from SQL prompt/developer
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3));
                count++;
            }
        }
    }
}
```

else

What is the difference b/w UpdatableRS and ReadOnlyRS?

Ans:- Using ReadOnly RS, we can perform only read Operations on the DB table.
Using Updatable RS, we can perform all CRUD Operations on DB table, that means
we can perform Insert,Update,Delete operations on DB table with out using Insert,Update,
Delete SQL queries.

What is the difference b/w UpdatableRS and SensitiveRS?

Ans:- while working with Updatable RS, the modifications done in RS object will reflect to DB table records...
where as while working with SensitiveRS, the modifications done in DB table records will reflect to RS object..

To create ReadOnly RS

```
Statement stmt=con.createStatement();
Statement st=con.createStatement() ResultSet.TYPE_SCROLL_SENSITIVE,
                           ResultSet.CONCUR_READ_ONLY);
ResultSet rs=stmt.executeQuery("SELECT * FROM STUDENT");
while(rs.next()){
    System.out.print(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3));
}
```

To create Updatable RS

```
Statement st1=con.createStatement();
Statement st=con.createStatement() ResultSet.TYPE_SCROLL_SENSITIVE,
                           ResultSet.CONCUR_UPDATABLE);
ResultSet rs1=st1.executeQuery("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
while(rs1.next()){
    System.out.print(rs1.getInt(1)+" "+rs1.getString(2)+" "+rs1.getFloat(3));
}
```

To read records [select operation]

```
while(rs.next()){
    System.out.print(rs.getInt(1));
}
```

To insert new record in DB table (Insert operation with out using Insert SQL query)

```
rs.moveToInsertRow(); //--> creates empty record in RS
rs.updateInt(1,800); // setting data to
rs.updateString(2,"Prashant"); // empty record of RS
rs.updateDate(3,now);
rs.updateFloat(4,78.88);
rs.insertRow(); // inserts new record to DB table
To update the record [without using update SQL query]
rs.absolute(1); // updates last value of 1st record
rs.updateString(2,"Amit");
rs.updateRow(); // this update reflects to DB table.
To delete the record
rs.deleteRow();
rs.absolute(1); // Delets 1st record from
rs.deleteRow(); // DB table and RS
```

Conclusion:- prefer using Insert,Update,Delete SQL queries directly to perform
non-select operations

What is the difference b/w Absolute PATH and relative PATH?

C:\abc

 |---xyz

 |---abc.txt

work location is C:\abc\abc Folder

```
absolute path of abc.txt :: e:\abc\abc\abc.txt (give from root directory)
relative path of abc.txt :: \abc\abc.txt (give w.r.t current location)
```

work location is E:\abc\123 folder

```
absolute path of abc.txt :: e:\abc\123\abc.txt (give from root directory)
relative path of abc.txt :: ..\abc\abc.txt
```

note:- If the PATH of the file given w.r.t current work location then it is called relative path and
this will change work location to work location...

note:- If the PATH of the file given right from root drive/directory then it is called absolute path and this will remain same from any work location.

How to make GUI Java App as desktop App? [Running app by using Icon]

Ans) yes we can know tool... `javaw` --> windows tool a built-in tool jdk 1.4/

step1] try to know Java version of the jar application ... [java 1.8]

step2] try to know the location package and class of the Application ...

(C:\Users\Admin\git\JAVA11\JDBCProj\bin)

RFC on Project ---> properties ---> advanced ---> location C:\Users\Admin\git\JAVA11\JDBCProj\bin

step3] make sure that JDBC driver jar file [jdbcs.jar] file is added to CLASSPATH.

step4] make sure that App Jdk version [java 1.8] is also in action mode at cmd prompt level... [java 1.8]

```
C:\Users\Admin\git\JAVA11\JDBCProj>java -version
java version "1.8.0_202-80"
Java(TM) SE Runtime Environment (build 18.0.2+14)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2+14, mixed mode, sharing)
```

step5] create short cut ...

RFC on desktop ---> short cut --->



Shortcut name = CollegeApp

step6] Add additional properties to the icon or short cut...

RFC on shortcut ---> properties --->

target : `"C:\Program Files\Java\jdk-18.0.2\bin\javaw.exe" com.mysql.ScrollFrame_GUIFE`

start in : `[C:\Users\Admin\git\JAVA11\JDBCProj\bin]`

short cut key : press any key then ctrlalt+key becomes short cut key

ctrl+alt+f4



notes: To work with javaw tool ... we need
jdk installation in computer...

step6] run the App by clicking on the icon

How to make Java CLI App as desktop App /CDN?

Ans) take the support of batch file... (.bat file)

notes: step1 to step 4 :- same as above

step5] create .bat file... on desktop

```
App.bat
=====
C:\Users\Admin\git\JAVA11\JDBCProj\bin
java com.mysql.ScrollFrame_GUIFE
pause
```

step6] run the batch file...

double click on the App.bat file..

How to create exe file representing Java GUI or CLI Application?

Ans) Run App ---> jar file ---> exe file

 ↳ JRE tool | Jar2Exe Tool - third party Tool

Need of creating exe file for Java App

- (a) To run Java App on non-Java Computer... (where jdk is not installed)
- (b) To pack and release the Project to Client Organizations...
- (c) To keep In the Internet as downloadable Application...
- and etc...

step1] create jar file representing our App having manifest file specifying the main class name...

 ↳ (jar file for holding additional info about jar file.)

 ↳ manifest.mf

 ↳ E:\desination\java_01\Java\bin

 ↳ manifest.mf

 ↳ --create

 ↳ --create_new_jar_file

 ↳ --specify_manifest_file

 ↳ --specify_jar_file_name

step2] download Jar2Exe tool from internet...



step3] Extract the s/w from zip file and install it...

step4] Launch Jar2Exe tool to convert given jar file to exe file...

 launch tool ---> next ---> browser and select the above jar file ---> choose java win[6] and mac[14] version
---> select windows ---> next ---> select GUI App ---> next ---> select splash lang --- and specify jar ---
---> select hide and encrypt class name ---> next ---> choose name and location of exe file ---> finish..

Working with different types JDBC drivers

There are 4/5 (5) mechanisms or architectures or methodologies to create JDBC driver s/w. for different DB s/w based on the rules and guidelines JDBC Technology...

- >Type1 JDBC driver [JDBC-ODBC Bridge Driver]
- >Type2 JDBC driver [Native API / Pure Java Driver]
- >Type3 JDBC driver [Net Protocol / All Java Driver]
- >Type4 JDBC driver [Native Protocol / All Java Driver]
- >Type5 JDBC driver [not officially given by Sun MI -- so there is no technical name for it]

- >> Vendor companies can give JDBC drivers for 1 or more DB s/w based on 1 or more architectures...
- >> Only sun MI has given Type1 architecture based JDBC Drivers upto Java 1.2... Now they stopped ... giving ... (Indirectly as of now there is no Type1 Architecture based JDBC driver in the latest Java versions)
- >> Based Type2 to Type5 architectures there are multiple vendor companies giving JDBC driver s/w's...

Applet

an applet is compiled java class that can be sent over the network as java based webpage.

two types Applets

a) Trusted Applets

>> These applets come to our computer from Internet and they allowed to interact/read/write operations with the file system of the computer. In that process it may bring virus computer (can damage computer).

>> These are not secured applets...

b) Untrusted Applets [By default all applets are untrusted Applets]

>> These applets come to our computer from Internet but can not interact with file system i.e can not perform read/write operations on the system... (no virus can be attached)

>> These are secured Applets...

Vendor DB Library / Native DB Interface

>> All JDBC drivers and some JDBC drivers (like type2) interacts and interacts with local or remote DB s/w with the support of Vendor DB Library...

>> In which case the Vendor DB Library comes in the form of .dll file | advanced exe file)

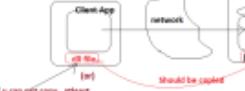
>> Once DB s/w is given, Vendor DB Library comes in form of .odbc.dll file...

>> In reality it comes as .msado15.dll...

>> In Client App to DB s/w communication, this dll file is required at client side...

note : If Client App and DB s/w are there in two diff computers .. we need arrange Vendor DB Library (.dll file) separately client computer...

Machine1 Client App .dll file .exe Machine2 DB s/w .dll file .exe

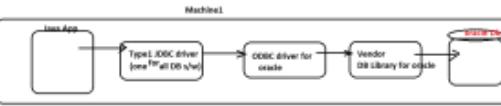


Type1 JDBC driver Architecture

↳ In this JDBC driver interacts directly with DB s/w by taking the support of JDBC driver and Vendor DB library

- >> Only sun MI has supplied this architecture based JDBC driver s/w as built-in JDBC driver for Jdk s/w (java.exe) ... Now not coding...
- >> By using single Type1 JDBC driver and different OSDB drivers and Vendor DB libraries , we can talk with multiple DB s/w's ... (for DB s/w's , single Type1 JDBC driver s/w is sufficient)

Machine1



Machine2



pros

>> One type1 jdbc driver can interact with multiple DB s/w's .. by using multiple Vendor DB Libraries and DB drivers

>> upto java7, It is built-in JDBC driver in Jdk s/w .. so there is no need arranging it separately...

>> Only for type1 architecture , sun MI is providing JDBC driver s/w's...

cons

>> Removed from Jdk 1.4 onwards.. and no other vendor is providing type1 JDBC driver s/w .. so simply outdated from Java 8

>> We need to arrange OSDB drivers and Vendor DB Libraries separately for each DB s/w's...

>> Since multiple compa are involved in DB s/w interaction, like OSDB drivers and Vendor DB libraries..

>> It is not industry standard JDBC driver s/w...

>> Does not support Untrusted Applets (DB s/w interaction)

>> It is a single thread driver (because methods are synchronized methods)... So it allows 1 thread at a time to interact with DB s/w through driver... (so performance is very bad)

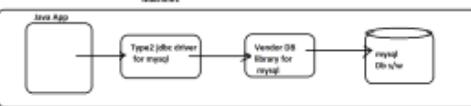
Why JDBC Driver is involved in type1 JDBC driver ?

- >> Initially Sun MI, thought of using already available OSDB drivers to interact with DB s/w's.. Since they failed to use directly... So they have given type1 jdbc driver using JDBC driver to interact with DB s/w's.

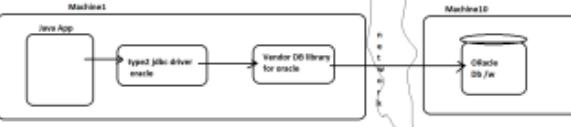
Type2 JDBC driver

>> It is given to interact with DB s/w directly by directly using Vendor DB library and without using the support of JDBC driver.

Machine1



Machine2



pros

>> no need of arranging DB drivers via s/w

>> Gives quite good performance when compare to Type1 JDBC driver s/w's

>> gives as multi-threaded JDBC driver

cons

>> This driver performance is not good for medium scale and large scale projects

>> Since vendor DB library is required at client machine.. This driver can not be used for untrusted Applet to DB s/w's communication

>> We need to arrange this driver s/w's from third party vendors...

>> For every DB s/w .. we need separate type2 JDBC driver s/w's...

e.g. type2 JDBC driver for oracle , type2 JDBC driver for mysql ...

>> It is not industry standard JDBC driver...

oracle corp supplied type2 jdbc driver for oracle is called oracle ocl driver (jcl->oracle call interface)

oracle corp supplied type2 jdbc driver for oracle is called oracle ocl driver

>>both these jdbc driver files are coming in the form of single jar file that is `ojdbc7.jar` or `ojdbc14.jar`

>>Type2/Type3 architecture based jdbc driver files are called "Thick JDBC drivers" becoz they need multiple copies to locate and interact with database ORA Uts.

>>Type4/Type5 architecture based jdbc driver files are called "Thin JDBC drivers" becoz they can locate and interact with Database DB Uts directly without taking the support of multiple other rungs like ODBC drivers and Vendor DB Libraries..

oracle ocl driver details (Oracle corp supplied type2 mechanism based jdbc driver for oracle)

jdbc driver class name:: oracle.jdbc.driver.OracleDriver [o] oracle.jdbc.OracleDriver

jdbc url :: `jdbc:oracle:thin:@`

protocol :: `thin`

sub name :: `system`

Jar file:: `ojdbc7.jar` (supports Auto-loading of jdbc driver class)

Example App:

step1) keep any oracle App ready
 step2) change file of `DriverManager.getConnection()`, as shown below..

```
Connection con=DriverManager.getConnection("abc:oracle:@abc","system","manager");
```

step3) run the App..

While running the above App, If the below exception comes

**Exception in thread "main" java.lang.UnsatisfiedLinkError:
 no oraclecl11 in java.library.path;** then we
 need to perform the following operations

[a] Add Oracle instantiable related: "bin" directory to PATH env., variable as the first value..

This path --> properties --> sys system settings --> env. variables --> system variables -->
 variable name :PATH
 add new value :: C:\javaweb\app\oracle\product\11.2.0\server\bin
 (refer image)

note :: Only few vendors are giving type2 JDBC drivers for DB Uts becoz
 Type2 are not industry standard..

Type4 JDBC driver

It is not official architecture to develop jdbc driver s/w...becoz architecture very much similar to type4 jdbc driver,
 but some third party vendors are giving it unofficially.

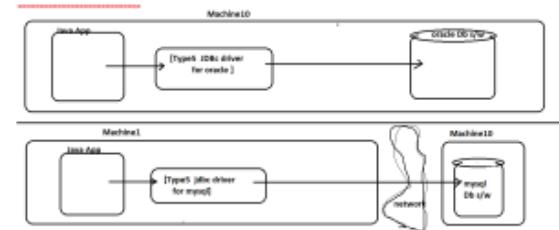
Advantages of type5 jdbc driver (Claimed by Vendor company)

- a) Consistent Performance..
- b) Flexibility of Enhancement (We can add new features easily)
- c) Standardization of Features .. | All JDBC features can be coded using common JDBC API interfaces... to work with different drivers and DB Uts..|
- d) All one deployment | everything comes in the form of single jar file|

and etc..

All these features are already found in Type2
 So there is no need of going for type5 drivers..

Type4 JDBC driver architecture



As of now "Progress DataDirect" company is giving Type5 JDBC drivers for different DB U/W ... as commercial drivers (30 days trial period)

Type5 driver for Oracle Details

driver class name:: com.ddtek.jdbc.oracle.OracleDriver
 JDBC url :: `jdbc:ddtek:oracle://localhost:1521/serviceName=orcl`
 protocol :: `thin`
 sub name :: `system`
 port number :: `1521`
 service id or logical DB :: `orcl`

Jar file :: `oracle.jar` [get it from <http://www.progress.com/download/interface/jdbc5/oracle/jdbc5oracle.jar>] using big Process

Example App:

d) keep any Type5 jdbc driver App ready

b) load driver class and establish the connection as shown below..

```
/register JDBC driver s/w (optional)  

Class.forName("com.ddtek.jdbc.oracle.OracleDriver");  

//establish the connection  

con=DriverManager.getConnection  

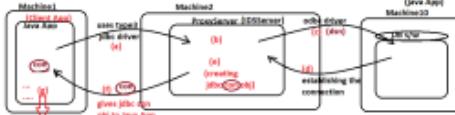
  ("jdbc:ddtek:oracle://localhost:1521/serviceName=orcl","system","manager");
```

c) add `oracle.jar` file to the build path of project.. (Selected from the group or right click on project)

RTC on Project --> build path --> click buildpath --> Libraries tab --> add external jars -->...

note: If u are using Faculty supplied `oracle.jar` file then change system date :: 2013 ..

d) run the application



(Here the Java App...
receives the JDBC connection object from the JDBC driver (a) to Java App.
and JDBC driver (b) to Java App.)
the JDBC driver takes the responsibility of creating JDBC connection object for multiple Client Apps/Java Apps. If any change in DB's location or details... will be taken care of proxy server... with out effecting the Client Apps/Java Apps.

Proxy Server means... the server that do work on behalf of another server/clients...

- > Windows installation gives built-in ODBC drivers ... like adobe driver for oracle, adobe driver for mysql, adobe driver for postgresql, adobe driver for ms-access and etc...
- >> every odbc driver is identified with its DSN (datasource name)
- 3 types of DSNs:
 - userDSN :: specific to current logged in windows user
 - System DSN :: specifies all the windows users of a computer
 - FileDSN :: Shareable in the network.

note: we need to create DSN by using some process...

iDServer

type: Proxy Server
Provider : Microsoft
default port no :: 12
Driver type: JDBC driver as built-in driver in <iDSERVER_HOME>/classes folder
connection s/w :: JDBC URL provider
url : <http://www.mkyong.com/download.html> (download free edition)

iDServer JDBC URL Definition

The URL format does not support the `DriverManager.getConnection(url,username,password)` method. Instead, it supports the `DriverManager.getConnectioon(url,username,password)` method.

eg: http://192.168.1.10:12/iDSERVER_HOME/testdb/testdb

>>Procedure to develop example App to Interact with Ms-Access DB s/w Using Type3 Driver supplied by iDServer

step[1] create LogInDB in Microsoft having DB table with records

Launch Ms Access (from MS-Office) --> new --> select DB name :: CollegeDB --> select location[C:\Workspaces\edhjava\N7A1414\College] -->

Open Tables in design view --> choose db table name (Student) --> add cols --> click --> enter values as records...



Windows file path: C:\Workspaces\edhjava\N7A1414\College\College.mdb

step[2] create System DSN In Oracle driver for Ms-Access...

search for oracle --> expand DSN tab --> add --> select MS Access Driver (*.mdb, *.accdb) -->

datasource name : access

DB --> select the above file G:\Workspaces\edhjava\N7A1414\College\College.mdb

Driver --> Oracle Database 11g Driver

Finish --> OK --> Next --> Next --> Finish



step[3] make sure iDServer is installed and running (After Installation it will start automatically)

step[4] Develop the Application using iDS server supplied type3 driver details

```
//register jdbc driver s/w (location)
Class.forName("iDServJDBCDriver"); //no auto loading of driver class
//establish the connection
con=DriverManager.getConnection("jdbc:iDServ://localhost:1234/testdb");
the above created DSN
```

step[5] make sure the C:\iDSERVER\classes folder is added to CLASSPATH or BuildPATH of Eclipse Project.

RTC on Project --> right click --> Properties tab --> JavaBuildPath --> add external class folder --> browse and select C:\iDSERVER\classes folder

refer Type3DriverSelectTest.java

step[6] run the Application..

try with resource (Java 7 feature)

=> So far we are using finally block to close JDBC obj and stream obj.
=> From Java 7 most of the objects are given auto closable objects.. i.e they will be closed automatically at certain point.
=> the class of the object is implementing java.lang.AutoCloseable(), then that object becomes autoclosable object automatically..(Java 7)
=> From Java 7 all stream objects are auto closable objects .. from Java 8 onwards ... JDBC obj are given as AutoCloseable obj.. The close() method of AutoCloseable interface executes .. automatically and closes the resource /object..

=> if we create object by using try with resource then .. at the end of try block the object will be closed automatically..

```
sync: try(resource {
  ...
  ...
} //At this Level the resource
catch{...} // will be closed automatically
}
catch{...}
}

note: resource object we are passing
try{} must be auto closable object..
```

```
tryConnection con=DriverManager.getConnection("..."); //either final variable or effectively final
try{...
  ...
  ...
} //here JDBC con object will be closed automatically
catch(SQLException self){...}
catch(Exception e){...} //we can not use con here
note: no need of taking finally block here..
```

Connection con=DriverManager.getConnection("..."); //either final variable or effectively final

```
try{...
  ...
  ...
} //here JDBC con object will be closed automatically
catch(SQLException self){...}
catch(Exception e){...} //we can not use con here
note: no need of taking finally block here..
```

try with resource in jdbc

=> its support is added to jdbc from jdk 1.8 ... [jdbc4] ...
=> All jdbc obj are auto closable ... So we can create/open them in try with resource.. and no need of closing them separately in finally block...becoz they will be closed automatically at the end of try block.

// Using java8 syntax of try with resource

```
try{Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "manager")){
    try{Statement st=con.createStatement();
        try{ResultSet rs=st.executeQuery("SELECT SNO,SNAME,SADD,AVG FROM STUDENT"){
            while(rs.next()){
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
            }
        }}while
    }//try3
} //try2
}//try1
catch(SQLException se){
    se.printStackTrace();
}
catch(Exception e){
    e.printStackTrace();
}
```

refer SelectTestUsingTWR.java

Java9 syntax based javac code having try with resource

```
try{Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "manager");
    Statement st=con.createStatement();
    try(st){
        ResultSet rs=st.executeQuery("SELECT SNO,SNAME,SADD,AVG FROM STUDENT");
        try(rs){
            while(rs.next()){
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
            }
        }}while
    } //try3
} //try2
}//try1
catch(SQLException se){
    se.printStackTrace();
}
catch(Exception e){
    e.printStackTrace();
}
```

refer SelectTestUsingTWR9.java

note:: While working with try with resource and other java8 features like LAMDA Expressions and etc.. we need to make most of the local variables either as final or as effectively final. So better not to declare local variables at the top of method definition having initial values while using the above features..

int age=0;
Calendar cal=null;  do not use from java 8 onwards...
Statement st=null;

Jdbc MetaData

MetaData :: data about data is meta data.. it may be gathering more data about existing data or it may be passing more data/info about existing data..

=> **MetaData/MetaInfo speaks about either gathering info or passing info about existing data/info.**

=> JDBC MetaData is about gathering more info about underlying DB s/w representing jdbc con obj... and underlying DB table represented by ResultSet object.. underlying query params[] represented by PreparedStatement or CallableStatement objs

3 types of JDBC MetaData::

- a) DatabaseMetaData (To know limitations and capabilities underlying DB s/w)
- b) ResultSetMetaData (To get More info Db table that is represented by ResultSet obj)
- c) ParameterMetaData (most of jdbc drivers not supporting this feature)

DatabaseMetaData

=> Gives limitations and capabilities of underlying Db s/w represented by jdbc con obj

=> To create this object

DatabaseMetaData dbmd=con.getMetaData();



we can call 50+ methods on this object to get more info underling DB s/w..

Some methods are::

getMaxColumnNameLength()	getMaxColumnsInGroupBy()
getMaxColumnsInIndex()	getMaxColumnsInOrderBy()
getMaxColumnsInSelect()	getMaxColumnsInTable()
getMaxConnections()	getMaxCursorNameLength()

refer DatabaseMetaDataTest.java

and etc..

=>DatabaseMetaData means It is the obj of underlying JDBC driver s/w supplied java class that implements java.sql.DatabaseMetaData[]()

note:: The values given by DatabaseMetaData object will change based on the DB s/w and jdbc driver we use..

=>if any method of JDBC MetaData programming is giving 0 or 0.0 or null .. that is not correct info about DB or table or param.. that indicates the underlying JDBC driver can not gather that info..

=>Realtime usecases of DatabaseMetaData ::

->Will be used in the development of GUI Db tools like SQL developer, mysql workbench ,sql yog and etc..

ResultSetMetaData

=> Gives more info about Db table like [col names, col types other details about cols] represented JDBC ResultSet object
=>ResultSetMetaData means it is the object of java class that implements java.sql.ResultSetMetaData().
=>To create ResultSetMetaData object
 ResultSet rs=st.executeQuery("SELECT * FROM STUDENT");
 ResultSetMetaData rsmd=rs.getMetaData();
=> we can invoke 20+ methods on "rsmd" object to know more about Db table represented by ResultSet object (rs).

String	getCatalogName(int column)	Gets the designated column's table's catalog name.
String	getColumnName(int column)	Returns the fully-qualified name of the Java class whose instances are manufactured if the ResultSet.getObject is called to retrieve a value from the column.
int	getColumnCount()	Returns the number of columns in this resultset object.
int	getColumnDisplaySize(int column)	Indicates the designated column's normal maximum width in characters.
String	getColumnLabel(int column)	Gets the designated column's suggested title for use in printouts and displays.
String	getColumnTypeName(int column)	Get the designated column's name.
int	getColumnType(int column)	Retrieves the designated column's SQL type.
String	getColumnTypeName(int column)	Retrieves the designated column's database-specific type name.
int	getPrecision(int column)	Get the designated column's specified column size.
int	getScale(int column)	Gets the designated column's number of digits to right of the decimal point.

=> ResultSetMetaData is very useful while generating reports and while developing GUI DB tools.

Refer ResultSetMetaDataTable.java

BFR	rs ResultSet	ALR
101 * raja hyd 90.55		101 raja hyd 90.55
102 ravi vizag 91.55		102 ravi vizag 91.55
103 * suresh delhi 87.55		103 suresh delhi 87.55

//DBC Code to display records of db table keeping col names and data types

```
//create ResultSetMetaData obj
ResultSetMetaData rsmd=null;
if(rs!=null)
    rsmd=rs.getMetaData();

int count;
if(rsmd!=null) {
    counter=rsmd.getColumnCount();
    //print col names
    for(int i=1;i<counter;i++)
        System.out.print(rsmd.getColumnLabel(i)+" ");

    System.out.println();
    //print col data types
    for(int i=1;i<counter;i++)
        System.out.print(rsmd.getColumnTypeName(i)+" ");

    System.out.println();
    //avg
    avg---> number(10,2)
    precision: 10
    scale :: 2
    89.55
    total length :: precision (10)
    scale :: 2
}
//display all records
if(rs!=null) {
    while(rs.next()) {
        for(int i=1;i<counter;i++)
            System.out.print(rs.getString(i)+" ");
    }
    System.out.println(); //new line
}
//while

//Getting more info about db table cols
for(int i=1;i<rsmd.getColumnCount();i++) {
    System.out.print("col index ::"+i);
    System.out.print("col name ::"+rsmd.getColumnLabel(i));
    System.out.print("col data type name ::"+rsmd.getColumnTypeName(i));
    System.out.print("col scale ::"+rsmd.getScale(i));
    System.out.print("col precision ::"+rsmd.getPrecision(i));
    System.out.print("col is nullable ? ::"+rsmd.isNullable(i));
    System.out.print("col is case-sensitive ? ::"+rsmd.isCaseSensitive(i));
    System.out.print("col is AutoIncrement ? ::"+rsmd.isAutoIncrement(i));
    System.out.println("-----");
}
//for
```

ParameterMetaData

=>Allows to gather more info about parameters (?) that in the SQL query represented by the PreparedStatement or CallableStatement object

=> Most of the JDBC drivers not supporting this concept.. becoz they are not providing any implementation class for this Interface "ParameterMetaData".

=> It is the object of underlying JDBC driver s/w supplied java class that implements java.sql.ParameterMetaData()

=>To create this object

```
PreparedStatement ps=con.prepareStatement("insert into Student values(?, ?, ?, ?);");
ParameterMetaData pmd=ps.getParameterMetaData();
```

=>we can invoke methods on "pmd" object to get more info parameters(...).

Modifier and Type	Method and Description	
String	getParameterClassName(int param)	Retrieves the fully-qualified name of the Java class whose instances should be passed to the method PreparedStatement.setObject.
int	getParameterCount()	Retrieves the number of parameters in the PreparedStatement object for which this ParameterMetaData contains information.
int	getParameterMode(int param)	Retrieves the designated parameter's mode.
int	getParameterType(int param)	Retrieves the designated parameter's SQL type.
String	getParameterTypeName(int param)	Retrieves the designated parameter's database-specific type name.
int	getPrecision(int param)	Retrieves the designated parameter's specified column size.
int	getScale(int param)	Retrieves the designated parameter's number of digits to right of the decimal point.
int	isNullable(int param)	Retrieves whether null values are allowed in the designated parameter.
boolean	isSigned(int param)	Retrieves whether values for the designated parameter can be signed numbers.

```
try{PreparedStatement ps=con.prepareStatement(GET_ALL_STUDENTS_QUERY){
    //create ParameterMetaData object
    ParameterMetaData pmd=null;
    if(ps!=null)
        pmd=ps.getParameterMetaData();
    //get more info about parameters
    int count=pmd.getParameterCount();
    System.out.println(count);
    for(int i=1;i<count;i++){
        System.out.println("parameter number::"+i);
        System.out.println("parameter mode::"+pmd.getParameterMode(i));
        System.out.println("parameter type name::"+pmd.getParameterTypeName(i));
        System.out.println("parameter scale::"+pmd.getScale(i));
        System.out.println("parameter precision::"+pmd.getPrecision(i));
    }
}
//try2
```

This code will give exception.. becoz All popular JDBC driver s/w are not supporting this ParameterMetaDAta... concepts..
`java.sql.SQLException: Unsupported feature`

Working with Properties file

=>The text file that maintains the entries in the form of key-value pairs is called properties file..

=>We can give any extension to properties file.. but the recommended extension is .properties

```
(com/nt/commons)
#Personal Information -comment
person.id=101
person.name=raja
person.addrs=hyd
```

java.util.Hashtable {any objs as keys, values,
 extends Map of elements}
java.util.Properties {Map collection}
(Only String objs as keys,values
 of elements)

props (obj of java.util.Properties)
person.id 101 0
person.name raja 1
person.addrs hyd 2
keys values
(String) (String)

Logic to copy Properties file info to java.util.Properties class obj

```
//Locate properties file
InputStream is=new FileInputStream("src/com/nt/commons/personalInfo.properties");
//create java.util.Properties object
Properties props=new Properties();
//Load from file
props.load(is); (Performs the actual task)

S.o.p(props);
S.o.p(" person.name key value is::"+props.getProperty("person.name")); //raja
S.o.p("person id ::"+props.getProperty("person.id")); // 101
```

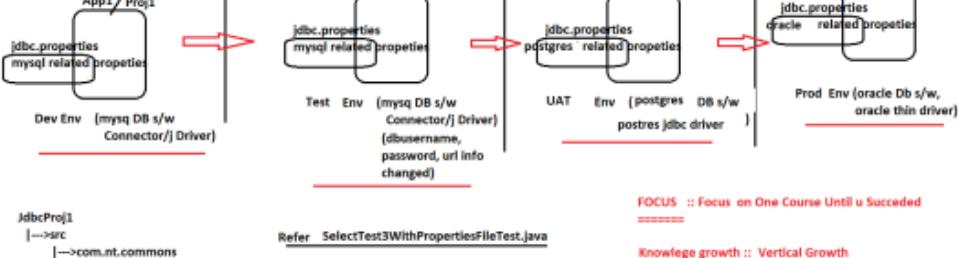
Collecting jdbc properties from Properties file

note:: The slogan in the software industry is . Do not hardcode inputs in ur App source code that are changeable in the future. Try collect such values to the Application from outside the Application.. by taking the support of different options like properties file or xml file or cmd line args or system properties or streams like Scanner and etc... (softcoding)

=> JDBC driver class name, JDBC url , db username, db pswd and etc.. are called JDBC properties. these values change based on JDBC driver and DB s/w we use.. So it is recommended not hardcode in our App source code.. collect them from outside the application using the one of the softcoding techniques..

(Generally we prefer properties file (standalone App) and XML file (web.xml) in web applications)

Use case:: => while moving the project from Dev Env.. to Test Env.. to UAT env ... Prod env.. The JDBC properties will change env.. to env.. becoz DB details will change env.. to env.. So better take the support of softcoding to supply JDBC properties to the Application.



FOCUS :: Focus on One Course Until u Succeeded



Example code

jdbc.properties (com/nt/commons)

```
# JDBC properties
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@localhost:1521:xe
db.user=system
db.pwd=manager

#jdbc.driver=com.mysql.cj.jdbc.Driver
#jdbc.url=jdbc:mysql://ntaj414db
#db.user=root
#db.pwd=root
```

In any JDBC App place code to collection JDBC properties from properties file and to create JDBC con object

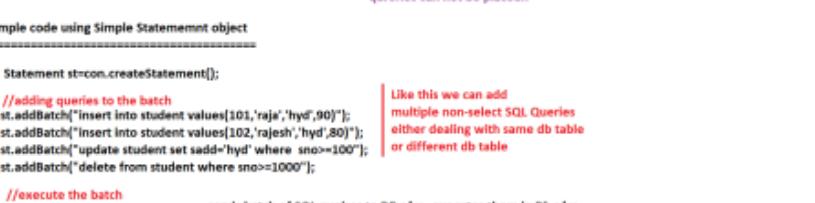
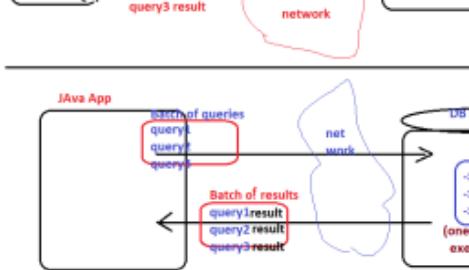
```
//Load the properties file into java.util.Properties class object
InputStream is=new FileInputStream("src/com/nt/commons/jdbc.properties");
Properties props=new Properties();
props.load(is);

//register JDBC driver s/w (optional)
Class.forName(props.getProperty("jdbc.driver"));
//establish the connection
Connection con=DriverManager.getConnection(props.getProperty("jdbc.url"),
                                             props.getProperty("db.user"),
                                             props.getProperty("db.pwd"));
```

Batch Processing /Batch Updation

=> In a normal JDBC App , to send and to execute "n" SQL Queries to DB s/w we need to use "n" roundtrips b/w Java App and DB s/w i.e we need to use network for 2*n times.. some it is very costly..

=> To overcome the above problem keep multiple related SQL queries in a batch and send batch to DB s/w.. execute the queries batch in DB s/w.. and gather query results as batch of results back to DB s/w .. It just uses one network roundtrip (2* 1 =2 times network) to dealwith batch of "n" SQL queries..



Sample code using Simple Statement object

```
Statement st=con.createStatement();

//adding queries to the batch
st.addBatch("insert into student values(101,'raja','hyd',90)");
st.addBatch("insert into student values(102,'rajesh','hyd',80)");
st.addBatch("update student set sadd='hyd' where sno>=100");
st.addBatch("delete from student where sno>=1000");

//execute the batch
int result[] = st.executeBatch(); // sends batch of SQL queries to DB s/w , executes them in DB s/w bring their results as batch in the form of int[] (result[])

//process the results
int total=0;
for(int i=0;i<result.length;i++){
    total+=result[i];
}
System.out.println("no.of records that are effected is "+total); //gives 5
```

Like this we can add multiple non-select SQL Queries either dealing with same db table or different db table

BatchProcessing does not perform "do every thing or nothing principle" based Query execution.. if want to apply that then go for TxMgmt on the top of batch results.

Can u explain different execute() methods of JDBC API?

Ans) executeQuery() :: For select SQL queries
executeUpdate() :: (For non-select SQL queries)
executeLargeUpdate() ::
execute() :: For both select and non-select Quies and also for calling PL/SQL procedures/functions
executeBatch() :: For batch processing.

Q) Why we can not add "select SQL queries to batch of Batch Processing"

Ans] The return type of executeBatch() method is int[] i.e it can not hold the select SQL query generated RS object .. So we can not add select SQL queries to batch of the BatchProcessing/Updation

So then how to execute group of select queries?

Ans] Not possible....

TransactionManagement

=>The Process of combining related operations into single unit and executing them by applying do everything or nothing principle is called TxMgmt(Transaction Management)

usecases:: 1. transferMoney operations (withdraw amt from source account, deposit amount into dest account) needs Txmgt
2. Employee Registration(insert record in hr table , insert record finance Db table) needs Tx mgmt

=>Batch Processing does not support TxMgmt by default .. we need to add additional logics on the top BatchProcessing

a) disable auto commit on Db s/w (con.setAutoCommit(false)) -->default true..

b) Write additional logic on the result[] of executeBatch to commit or rollback Tx

```
boolean flag=false;
for(int i=0;i<result.length;i++){
    if(result[i]==0){
        flag=true;
        break;
    }
}
if(flag==true){
    con.rollback(); //Rollback Tx //TCL
}
else{
    con.commit(); //commit Tx //TCL
}
```

Why commit() or rollback() are given on "con" object, why not in Statement object?

Ans] Since we execute different SQL Queries using different Statement objs ..In order to commit or rollback all these queries related TxMgmt in single shot they have given con.commit() and con.rollback() methods in connection obj who creates multiple statement objects.

TxMgmt Code

a) Begin Tx
con.setAutoCommit(false);

b) execute logics/queries
=>we can go for batch processing /updation

c) commit or rollback Tx

=>based on the results of batch processing/updation
we can call con.commit() /con.rollback() method

TransferMoney Operation Having JDBC TxMgmt

Db table (JDBC_ACCOUNT)		
acno(pk)	holdername	balance
101	raja	90000
102	rani	80000

refer TxMgmt_TransferMoneyTest.java

=>Do not use try with resource while working TxMgmt becoz.. doing con.commit() , con.rollback() inside try block is not good practice. It is good to do in finally block .. so prefer using the traditional try- catch -finally clauses in Tx Mgmt...

```
try{
    ...
    ...
    ...
}
catch(Exception e){
    ...
}
finally{
    check cond
    //con.commit() or con.rollback()
}

=> finally not possible while working
with try with resource...
=> In try with resource con obj will be closed at
the end of try block.. So calling con.commit() or con.rollback()
throws exception.

try(Connection con=DM.getConnection(...)){
    ...
    ...
    ...
}
catch(Exception e){
    ...
}
//con.commit() or con.rollback() not possible here.
```

Not at all good

List<String> list=new ArrayList() ; //bad //gives mutable object

List<String> list= List.of("A","B","C"); //good //give immutable object // from java9

~~(Not best in all cases)~~

(Q) In generally why we close all connections after use... what is the main reason behind closing?
Ans] To avoid memory leak(s) we should close connection and streams at the end of their utilization.

About: Memory Leaks in Java
<https://www.java-tutorial.net/memory-leaks-java/>

Note: We can not use PreparedStatement object in batch processing towards adding multiple SQL queries to that batch; hence each PreparedStatement object can represent only one pre-compiled SQL query; same limitation is also there with CallableStatement objects.

PreparedStatement ps
ps = connection.prepareStatement("UPDATE JCMB_BANKACCOUNT SET BALANCE=BALANCE+1 WHERE ACNO=?");

Interacting with unconventional DB's/webs

- 3 types of DB's/webs:
 - [---] Conventional DBMS (DB): DB's like eg: oracle, mysql, postgresql, and etc..
 - [---] UnConventional DB's/webs:
 - eg: Ms Excel, TextFiles (xml files, Table file) [These are not DB's/webs.. but we are going to take them as DB's/webs]
 - eg: MongoDB, Cassandra and etc..

What is the need of unconventional DB's/webs?

Advantages over conventional DB's/webs as main DB's/webs.. and unconventional DB's/webs.. for taking point notes.. for managing data temporary..

Disadvantages: JDBC don't aware about DB's/webs.. for managing DB's/webs.. and uses JDBC driver as helper DB's/webs.. for account statement purpose other every known.

Example: The teachers' collected status info will be typed to Ms-Excel by JDBC (data entry operation) and later it will be copied To Main DB's/webs like Oracle.

<<HOTT Company is supplying Jdbc driver based JDBC driver for Ms Excel and JDBC driver Test.xlsx..

URL: <http://www.hott.com/test.xlsx.html>

Interacting with Ms-Excel

Step1) Create Microsoft Word Book and sheets with Data.. and Inserting value in Logical DB, db table



Step2] Arrange HOTT company supplied Excel JDBC drivers.. and gather its details

Jdbc driver class name: com.hott.jdbc.ExcelDriver

url: jdbc:Excel:///Directory location where xlsx file is saved

jar file: Excel JDBC.jar (supports auto loading of JDBC driver class)

Note: we can download entire driver jar as zip file from: <http://www.hott.com/test.xlsx.html>

Step3] Develop the Application using HOTT JDBC driver

refer [Result.xlsxTest.java](#)

(2) In generally why we close all connections after use... what is the main reason behind closing?

An: To avoid memory leak; we should close sockets and streams at the end of their utilization.

About: Memory Leaks in Java
https://www.tutorialspoint.com/memory_leaks_in_java.htm

Note: We can not use PreparedStatement object in batch processing towards adding multiple SQL queries to that batch;

because each PreparedStatement object can represent only one pre-compiled SQL query; same limitation is also there with CallableStatement object.

PreparedStatement ps;

ps = connection.prepareStatement("UPDATE JCBC_BANKACCOUNT SET BALANCE=BALANCE+1 WHERE ACNO=1");

Interacting with unconventional DB's/webs

3 types of DB's/webs:
1. Conventional DBMS (DB): DB's like eg: Oracle, MySQL, PostgreSQL, and etc..
2. Unconventional DB's/webs:
a. Web based - TestFiles (.jar files, Table file) [These are not DB's/webs.. but we are going to take them as DB's/webs]
b. NoSQL - MongoDB, Cassandra and etc..

What is the need of Unconventional DB's/webs?

Advantages over conventional DB's/webs are main DB's/webs.. and unconventional DB's/webs.. for taking point notes.. for managing data temporary.

Disadvantages: JDBC does not work as main DB's/webs.. for managing DB's/webs.. and uses JDBC as helper DB's/webs.. for account statement purposes other every known.

Disadvantages: The tables' collected status info will be typed to MySQL by JDBC [Data Entry operation] and later it will be copied.

To Make DB's/webs like Oracle.

<<HOTT Company is supplying Jdbc driver manager based JDBC driver for MS Excel and JDBC driver Test Files..

URL: <http://www.hott.com/jdbc.html>

<http://www.hott.com/testdb.html>

Interacting with Ms-Excel

step[1] Create Microsoft Word book and sheets with Data.. and Industry when in Logical DB.. db table

and physical DB in that

1/Imports	Microsoft Word	Microsoft Office [Physical DB]	1/Sheets	Jobs table[1]
[-->import]	[-->import]	[-->import]	[-->sheet]	[-->sheet]
[-->import]	[-->import]	[-->import]	[-->sheet]	[-->sheet]
[-->import]	[-->import]	[-->import]	[-->sheet]	[-->sheet]
[-->import]	[-->import]	[-->import]	[-->sheet]	[-->sheet]
[-->import]	[-->import]	[-->import]	[-->sheet]	[-->sheet]

step[2] Arrange HOTT company supplied Excel JDBC drivers.. and gather its details

Jdbc driver class name: com.hott.jdbc.ExcelDriver
url: jdbc:ExcelDriver://Directory location where .xls file is saved

jar file: Test_JDBC01.jar [Imports auto loading of JDBC driver class]

note: we can download entire driver jar as zip file from: <http://www.hott.com/testdb.html>

step[3] Develop the Application using HOTT JDBC driver

refer [ExcelTest.java](#)

note: Excel_JDBC01.jar files must be added build path of the project

public class ExcelTest{

```
    //Connection connection=DriverManager.getConnection("jdbc:ExcelDriver://D:\\Workspaces\\WebServices\\NTA414\\");
    private static final String EXCEL_ALL_SELECT="SELECT * FROM COLLEGE.SHEET1";
    public static void main(String[] args) {
        try(Connection conn=DriverManager.getConnection("jdbc:ExcelDriver://D:\\Workspaces\\WebServices\\NTA414")){
            try(PreparedStatement ps=conn.prepareStatement(EXCEL_ALL_SELECT)){
                try(ResultSet rs=ps.executeQuery()){
                    while(rs.next()){
                        System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
                    }
                }
            }
        }catch(SQLException se){
            se.printStackTrace();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

1/1/main

For inserting records to Excel01.. refer [ExcelTest.java](#)

Interacting with Text files (CSV files) using JDBC

step[1] create .csv file in any folder of c:/choice

1/1/Workspaces	Logical DB	file1.csv
physical	[-->import]	[-->import]
DB	[-->import]	[-->import]
	[-->file1 CSV (db table)]	181,java,100,00,07
	[-->file1 CSV (db table)]	182,java,100,58,32
	[-->file1 CSV (db table)]	183,java,100,04,74
	[-->file1 CSV (db table)]	184,java,100,05,19

step[2] Gather details.. And required Test driver

Jdbc driver class name: com.ibm.db2.jcc.DB2Driver

jar file: jdbc:DB2://logicalDB - setting test location of csv file:

jar file: Test_JDBC01.jar [Imports auto loading of JDBC driver class]

note: here file on file names [like file1.csv / file2.csv will be taken as db-table names]

note: Test_JDBC01.jar files must be added build path of the project

step[3] Develop the Application....

CSVTest.java

public class CSVTest{

```
    //Connection connection=DriverManager.getConnection("jdbc:DB2://");
    private static final String CSV_ALL_SELECT="SELECT * FROM file1";
    public static void main(String[] args) {
        try(Connection conn=DriverManager.getConnection("jdbc:DB2://")){
            try(PreparedStatement ps=conn.prepareStatement(CSV_ALL_SELECT)){
                try(ResultSet rs=ps.executeQuery()){
                    while(rs.next()){
                        System.out.println(rs.getString(1)+" "+rs.getString(2)+" "+rs.getFloat(3));
                    }
                }
            }
        }catch(SQLException se){
            se.printStackTrace();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

1/1/main

For inserting records into CSV file refer: [CSVTest.java](#)

Select SQL Query is giving 50 records but I want see only 10 records. In to - its object with changing SQL query?

An: rs.setFetchSize(10); or rs.setFetchSize(10); makes limit of JDBC drivers are not supporting this features..

RowSets

==> Rowset is extension of ResultSet to hold bunch of records given by "select SQL query" execution.

==> Every RowSet object is the object of java class that implements javax.sql.RowSet[]



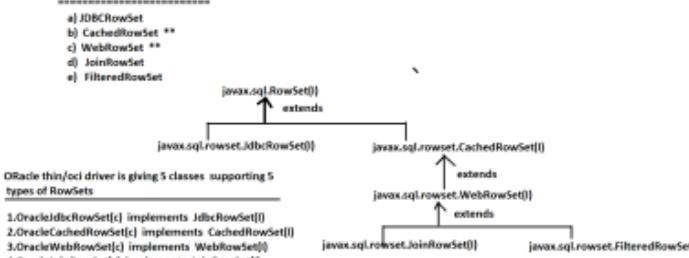
Limitations of RowSet

- [a] RowSet objects are connected objects i.e we can not do offline Data manipulation with out having connectivity with Db s/w.
- [b] Does not support Bean style Programming (Programming with setter/getter methods)
- [c] Does not support event handing
- [d] RowSet obj are not Serializable objects i.e we can not send them over the network..

To overcome these limitations we can use RowSets and advantages are

- [a] we can get all basic functionality of ResultSet object
- [b] ** these disconnected objects i.e we can modify RowSets objects in offline mode with out having connection with Db s/w... and we make these modifications reflecting to Db s/w once Connection is back
- [c] supports bean style programming
- [d] Supports event handing. [The action done one record of RowSet can be made reflecting on other records]
- [e] Most of the RowSet obj are Serializable obj, So they can be sent over the network..

5 types of RowSets are there



ORACLE thin/oci driver is giving 5 classes supporting 5 types of RowSets

1. OracleJdbcRowSet[c] implements JdbcRowSet[]
2. OracleCachedRowSet[c] implements CachedRowSet[]
3. OracleWebRowSet[c] implements WebRowSet[]
4. OracleJoinRowSet[c] implements JoinRowSet[]
5. OracleFilteredRowSet[c] implements FilteredRowSet[]

JdbcRowSet

- ==> It is connected RowSet object
- ==> It is not a Serializable object
- ==> It is given in RowSet to work like ResultSet
- ==> Supports Bean Style programming
- ==> Does not support offline data manipulation
- ==> These Rowsets are default scrollable
- ==> we can use Sensitive or Inensitive Behaviour JdbcRowSet..
- ==> These RowSets are Read Only Rowsets by default and we can not make them as Updatable Rowsets in any angle

Q) How restrict no.of records That should come RS or RowSet though its select SQL query is giving more records?

Ans] rs.setFetchSize(n); or rowset.setFetchSize[] methods are not working in most of the JDBC drivers.. as alternate we can use st.setMaxRows(n) or rowset.setMaxRows[n] methods.

```

try{OracleDBRowSet jrowset=new OracleDBRowSet();
//set db properties
jrowset.setURL("jdbc:oracle:thin:@localhost:1521:xe");
jrowset.setUsername("system");
jrowset.setPassword("manager");
//set SQL Query
jrowset.setCommand("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
jrowset.setMaxRows(4);
//execute the SQL query
jrowset.execute();
//process the RowSet
jrowset.setReadOnly(false);
while(jrowset.next()){
    //process the RowSet
    jrowset.acceptChanges();
    Thread.sleep(10000);
    Thread.sleep(10000);
}
System.out.println(jrowset.getInt(1)+" "+jrowset.getString(2)+" "+jrowset.getString(3)+" "+jrowset.getFloat(4));
}
}
jrowset.absolute(1);
System.out.println("-----Stop DB s/w-----");
jrowset.updateString(3,"newyork");
jrowset.updateRow(); //This is called offline/disconnected data processing
System.out.println("-----Start DB s/w-----");
Thread.sleep(40000);
//process the RowSet
jrowset.acceptChanges();
while(jrowset.next()){
    System.out.println(jrowset.getInt(1)+" "+jrowset.getString(2)+" "+jrowset.getString(3)+" "+jrowset.getFloat(4));
}
}
}
  
```

[refer OracleRowSetTest.java](#)

CachedRowSet

- ==> It is disconnected RowSet i.e offline processing is possible
- ==> It is Serializable object
- ==> It is Sensitive/Inensitive. Updatable/Readonly (Both read and write are possible)
- ==> The modifications done in offline env.. (not having connection with DB) can be reflected back to DB after getting connection by calling acceptChanges() method.

==> While developing PDA [Personal Digital Assistant] ->like tab applications these are very popular.

e.g.: Flipkart delivery boy tab, Insurance agent Tab , Bank Marketing Employee tab and etc..

CachedRowSetTest.java

```

try{OracleCachedRowSet crowset=new OracleCachedRowSet();
//set db properties
crowset.setURL("jdbc:oracle:thin:@localhost:1521:xe");
crowset.setUsername("system");
crowset.setPassword("manager");
//set SQL Query
crowset.setCommand("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
// crowset.setMaxRows(4);
//execute the SQL query
crowset.execute();
//process the RowSet
crowset.setReadOnly(false);
while(crowset.next()){
    System.out.println(crowset.getInt(1)+" "+crowset.getString(2)+" "+crowset.getString(3)+" "+crowset.getFloat(4));
}
System.out.println("-----Stop DB s/w-----");
Thread.sleep(40000);
crowset.absolute(1);
crowset.updateString(3,"newyork");
crowset.updateRow(); //This is called offline/disconnected data processing
System.out.println("-----Start DB s/w-----");
Thread.sleep(40000);
//process the RowSet
crowset.acceptChanges();
while(crowset.next()){
    System.out.println(crowset.getInt(1)+" "+crowset.getString(2)+" "+crowset.getString(3)+" "+crowset.getFloat(4));
}
}
}
  
```

[refer CachedRowSetTest.java](#)

WebRowSet

==> same as Cached Rowset , just very useful to render DB table data as xml data

and also to write xml file.

==> Very useful in Soap Based WebServices where data will be carried in the form of Xml

==> Useful in Ajax Extension of JS env.. to send to response browser as xml content..

Note: XML is a mechanism of coding, describing data having branch rules and we can also send this data over the s/w/text data having Platform and s/w Independence..

[refer WebRowSetTest.java](#)

```

try{OracleWebRowSet wrowset=new OracleWebRowSet();
//set dbc properties
wrowset.setURL("jdbc:oracle:thin:@localhost:1521:xe");
wrowset.setUsername("system");
wrowset.setPassword("manager");
//set SQL Query
wrowset.setCommand("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
// wrowset.setMaxRows(4);
//execute the SQL query
wrowset.execute();
//process the RowSet
wrowset.setReadOnly(false);
while(wrowset.next()){
    System.out.println(wrowset.getInt(1)+" "+wrowset.getString(2)+" "+wrowset.getString(3)+" "+wrowset.getFloat(4));
}
System.out.println("Writing RowSet object data to XML file");
//write to xml file
Writer writer=new FileWriter("student_info.xml");
wrowset.writeXml(writer);
System.out.println("Writing RowSet object data on the console in xml format");
wrowset.writeXml(System.out);
}
}
  
```

[refer OracleWebRowSetTest.java](#)

```

JoinRowSet
=====
=>Allows to combine multiple Rowsets into single Rowset by applying SQL inner joins
try{OracleCachedRowSet crs1=new OracleCachedRowSet();
    OracleCachedRowSet crs2=new OracleCachedRowSet();
    OracleJoinRowSet jRowSet=new OracleJoinRowSet();
    crs1.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
    crs1.setUsername("system");
    crs1.setPassword("manager");
    crs1.setMatchColumn(1);
    crs1.setCommand("SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP");
    crs1.execute();

    crs2.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
    crs2.setUsername("system");
    crs2.setPassword("manager");
    crs2.setMatchColumn(1);
    crs2.setCommand("SELECT DEPTNO,DNAME,LOC FROM DEPT");
    crs2.execute(); refer JoinRowSetTest.java

    //add multiple cached rowsets to joinrowsets
    jRowSet.addRowSet(crs2);
    jRowSet.addRowSet(crs1);

    //process the joinrowset
    while(jRowSet.next()) {
        System.out.println(jRowSet.getString(1)+" "+jRowSet.getString(2)+" "+jRowSet.getString(3)
            +" "+jRowSet.getString(4)+" "+jRowSet.getString(5)+" "
            +" "+jRowSet.getString(6)+" "+jRowSet.getString(7));
    }

}
catch(SQLException se) {
    se.printStackTrace();
}
}

```

Filtered RowSet

=> Allows to filter the records that should come to RowSet based on java code based conditions..
=> This Condition will be prepared as Predicate object nothing but class that is implementing Predicate() and we write conditional logic in the evaluate() method.
=> For every record this evaluate() method executes.. if this method returns true then record will be there in FilteredRowSet otherwise will not be there in filteredRowSet.

Example code

```

try{OracleFilteredRowSet frs=new OracleFilteredRowSet();
    frs.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
    frs.setUsername("system");
    frs.setPassword("manager");
    frs.setCommand("SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP");
    frs.setFilter(new Filter("ENAME","S"));
    frs.execute();

    //process rowset
    while(frs.next()){
        System.out.println(frs.getInt(1)+" "+frs.getString(2)+" "+frs.getString(3)+" "+frs.getFloat(4)+" "+frs.getInt(5));
    }

}
catch(SQLException se) {
    se.printStackTrace();
}
}

private static class Filter implements Predicate{
    private String colName;
    private String condData;

    //alt +shift +s,o -> generates constructor
    public Filter(String colName, String condData) {
        this.colName = colName;
        this.condData = condData;
    }

    @Override
    public boolean evaluate(ResultSet rs) {
        System.out.println("FilteredRowSetDemo.Filter.evaluate()");
        try {
            String colValue=rs.getString(colName);
            if(colValue.startsWith(condData))
                return true;
            else
                return false;
        }
        catch(SQLException se) {
            //se.printStackTrace();
            return false;
        }
    }

    @Override
    public boolean evaluate(Object value, int column) throws SQLException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean evaluate(Object value, String columnName) throws SQLException {
        // TODO Auto-generated method stub
        return false;
    }

}
}

note: It is better condition directly in the SQL query itself instead working with FilteredRowset having condition java code..

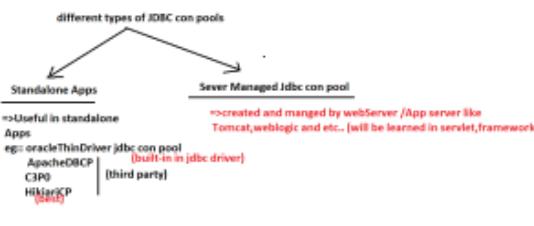
```

Flow :: when frs.next() method is called.. each record given Select Query of "frs" goes to evaluate() as RowSet object there read given "ename" col value and checks against given condition data and return true/false .. if true then it keeps the record in "frs" otherwise it will not keep. (frs->Filtered RowSet)

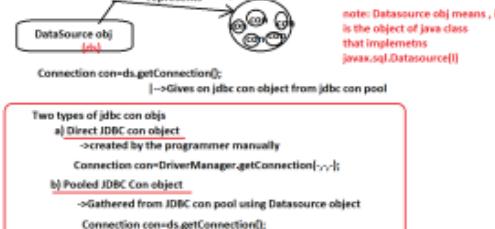
note:: Industry uses CachedRowSet and WebRowSet in projects..

JDBC Con pool

=> It is the factory that contains set of readily available jdbc con obj before actually being used..



=>DataSource obj acts entry point for jdbc con pool.. i.e to create con obj in jdbc con pool and to get con objects from jdbc con pool we need DataSource object.



- Two types of jdbc con obj
- a) Direct JDBC con object
 - >created by the programmer manually
 - Connection con=DriverManager.getConnection(...);
 - b) Pooled JDBC Con object
 - >Gathered from JDBC con pool using Datasource object
 - Connection con=ds.getConnection();

```

Example
=====
OracleDataSourceConnectionPool ds=new OracleConnectionPoolDataSource();
ds.setURL("jdbc:oracle:thin:@localhost:1521:xe");
ds.setUser("system");
ds.setPassword("manager"); //use all these details and creates
                           //jdbc con pool having initial 4 jdbc con obj (default)
//get pooled jdbc con obj
con=ds.getConnection();
//create Statement obj
Statement st=con.createStatement();
//Send and execute SQL query
ResultSet rs=st.executeQuery("SELECT SNO,SNAME,SADD,Avg FROM STUDENT");
//process the RS
while(rs.next()){
    System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
}

```