Before adv.java course (u r just web user/ website visitor)
after adv.java course (u will be called web Developer /website developer)

Main topics of our course
=========================
JDBC and misc
Jndi and misc
Servlet and misc
Jsp and misc
Html and Java Script
Tools (maven/gradle, log4j ,junit, GIT, Agile-JIRA and etc.. 20+....)
MiniProjects
Design Patterns (6 to 7)

=>Java to DB connectivity
=>Website Development
=>website hosting
and etc...

Adv.java
pre-requisite: core java oops
duration :: 100 ssessions
daily 1.30 mins to 1.45 mins (weekly 6/7 sessions)
Course Fee: Rs.3500/-
Admin :: srikanth
9302968665 (Only whatsaup texting)

FB group :: nataraj avaneen
https://www.facebook.com/groups/388099825162910
email id :: natarajavaarene@gmail.com

=>Since java can be used to develop various types of apps ,we can java as programming suite

Important modules of Java Programmign suite

a) JSE module
b) JEE module
c) IME module

JSE module
==========
JSE ::JAva Standard Edition
=>It is base module for other module
=> It is installable s/w as jdk s/w
=>Latest version :: java18
stable versions in the industry javaS , 11
=>This module given to develop Standalone Apps , Applets (outdated)

standalone Apps
===============
=>The App that is specific one computer and allows only user a time to operate the app is called
called standalone App

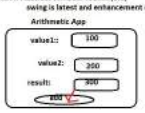
=>These are not user-friendly becoz every thing will be in the for text/chars .. no graphical user interface..
=> The basic core java App that is having main(-) in the class is called CUI App/Console App

=>These are user-friendly Apps becoz It will provide Graphical user interface for enduser to operate the Application like Buttons ,text boxes and etc..

=> Java Awt /Swing Apps are Desktop/GUI Applications.

Awt :: Abstract Winslow Toolkit (old)
swing is latest and enhancement of awt

TestApp1.java
```
public class TestApp1{
  public static void main(String args[]){
    ....
    ....
    ...
  }
}
```

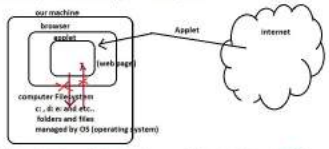


These Apps also contain
class with main(-) method ..
But uses AWT/Swing concepts
in order provide Graphical User Interface

Applets
=======
=> Applet is a compiled java class that can be sent over the network as java based web page

=> Html web pages are good in performence , but very bad security..
=> Applet based webpages good in security and but bad in performence (takes more time to load on to browser)

=> By default all applets untrusted Apps .. then are coming to our computer from internet as web pages
then can not write anything to computer and they can not read anything from computer.. So applets can not bring virus to our computer .This indicates applets secured by default



=>Applets are outdated becoz of their bad performence .. As of now industry is using HTML to to develop the webpages but improving security standards with support other java java technologies like servlet ,jsp and etc.. By also using networking concepts like firewalls.

JSE module consits two parts
a)part1 :: Java language
b) part2 :: Basic Java Technologies (JDBC,INDI, RMI and etc..)

Java Java language concepts in JSE module are
=> oops, exception handling, collections, multi-threading, IO streams, networking,
Reflection api , AWT ,Swings and etc..
=> Java technologies of JSE modules are (Trade mark)
JDBC (Java DB connectivity) JDBC -->TM (no abbriviation) (Latest)
(old days)
JNDI (Java Naming Directory Interface)
RMI (Remote Method Invocation)
and etc..

Core java course :: deals with language part of JSE module (java based standalone App devleopment)
Adv.java course :: deals with JSE module technologies (JDBC,JNDI) and JEE module technologies (Servlet,jsp, java mail and etc..)
[All these things generally we use in java based web site development]

=> JAva Learning is all about learning
a) java language (core Java) (like raw materials like rice, wheat granuals)
b) java technologies (adv.java) (semi-finished products like rice powder, wheat powder)
c) Frameworks (spring, hibernate, webservices and etc..) (Fully finished product like Maggie noodles/bake noodles)

=>TIME old days no abbriviation
=> T.I.M.E (Triumphant Institute of Management Education)

In resume
Programming languages :: java , c
JSE Technologies :: JDBC, Jndi
JEE Technologies :: Servlet,jsp, java mail
UI Technologies :: Html, java script , css, bootstrap ,..

Before adv.java course (u r just web user/ website visitor)
after adv.java course (u will be called web Developer /website developer)

Main topics of our course
==========
JDBC and misc
(ndi and misc
Servlet and misc
Jsp and misc
Html and Java Script
Tools (maven/gradle, log4j ,junit, GIT, Agile-ERA and etc.. 20+...)
MiniProjects
Design Patterns (6 to 7)

=>Java to DB connectivity
=>Website Development
=>website hosting
and etc...

FB group :: nataraj javaarean
https://www.facebook.com/groups/388095825162910
email id :: nataraj javaarena@gmail.com

Adv.java
pre-requisite:: core java oops
duration :: 100 +sessions
daily 1.30 mins to 1.45 mins (weekly 6/7 sessions)
Course Fee: Rs.3500/-
Admin :: srikanth
6302968665 (Only whatsup texting)

Why did u choose Java ?

=>Java is opensource
(free and java source code is visible to every one)
=> Java apps are WORA (Write Once Run AnyWhere) (portable across the multiple Operating systems)
.java ----> .class (win) .class (linux) .class (unix) .class (mac)
(This is also called Platform independent nature)

=> Java can be used to develop different types of Apps
like standalone Apps, websites, mobile Apps (andriod Apps), Distributed Apps
BigData procesing , DataScience and etc.., Jtesting (Like GPay App)
(FB Data processing) (35+ technologies) (Selenium)
(Hadoop, Spark)

python is good
In DataScience (AI,ML,DL,...)
and IOT
(Internet of Things)
(Home Automation)

The devices and non-living things will use internet directly ...

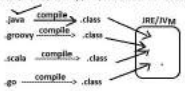
=> Java is secured
-->.class tampering is not allowed.. if we do , the execution will fail
--> UnTrusted Applets can not write and read to /from computer
--> we can keep web comps (the prgs generating web pages) in private areas to hide their source code (source code protection is possible)
->While delivering we can deliver only .class files for hiding the source code..

(source code)
.java ----javac(compiler)----> .class (byte code)
.class ----> decompiler ----> .java file (source code)
(byte code) (No decompiler is 100% accurate)

=> Java is stable and can be used long run projects..

=> ofcourse java domain is having more openings..

Java , koltin , Groovy , Go ,scala and etc.. JVM basaed languages..



=. C#.net , D2k, VB , PB and etc.. are compitetors for JSE module to devleop standalone App but their Apps platform dependent Apps where JSE based Java Apps are platform indepedent Apps

D2k -->Developer 2000
VB -->Visual Basic
PB --> Power Builder

all these apps are non-WORA Applicaitons where JSE module Apps WORA Applications..

youtube vedios
==========
https://www.youtube.com/watch?v=2oTIHv-GwII
https://www.youtube.com/watch?v=eiu2eXseCCU&list=PLVICJHNRLfIP9OITKTQuq3UWINA_wOPIFr

Adv.Java
pre-requisite:: Core Java oops
duration is : 100+ sessions
daily 1:30 to 1.45 sessions
weekly 6/7 sessions
Course fee :: rs. 3500/-
Admin :: srikanth
contatct no : +91 6302 968 665 (only whatsup)

Backup videos ::
https://www.youtube.com/watch?v=2oTiHv-GwtI&t=55s
class notes ::
https://www.facebook.com/groups/388095825162910/files
FB group name:: nataraajavaarena
email id :: nataraajavaarena@gmail.com

Important modules in java are
a) JSE (Java standard Edition)
b) JEE (Java Enterprise Edition / Jakartha enterprise Edition)
c) JME (Jvaa Micro /Mobile Edition)

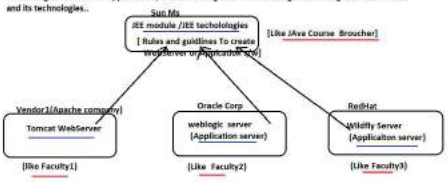
JEE module
==========
JEE :: JAva Enterprise Edition /Jarkatha Enterprise Edition
Latest version :: JEE 8
It is not installable s/w , It contains set of s/w technologies/specifications providing rules and guidelines to create WebServer(like Tomcat) or Application Server(like weblogic) s/w.

Application server =webServer ++

=>Working with these webServer or Application server s/ws is nothing but working with JEE module and its technologies
=> JEE module technologies are Servlet,Jsp, EJB, Java mail , JMS , JTA and etc..

Jsp :: Java server Pages
EJB :: Enterprise Java Beans
JMS :: Java Messaging Service
JTA :: Java Transaction API

=>Installing WebServer or Application s/w and working with it .. It nothing but working with JEE module and its technologies..



In order work JEE module or its JEE Technologies , we never arrange them becoz the they are not installable s/ws, we arrange JEE module/Technologies based webSErver/Applicalton server s/w (like Tomcat or weblogic or Wildfly or etc..) and we use that server s/w to devleop the software Apps.

=>Working with webServeror Application server s/w and using them in software App devleopment is nothing but working with JEE module and its technologies (It like attending Faculty conducted course is nothing reading the course topics that are defined in Course Broucher)

=> The s/w company who creates and releases s/ws to market is called Software Vendor company
a) Sun Ms b) IBM c) Apache d) oracle corp e) MicroSoft e) Eclipse Org f) Google and etc..

=>The s/w company who uses the s/w vendor supplied softwares to develop software projects is called software services company
eg:: Wipro , Infosys , iGate, CapGemini , TCS , CTS , HCL and etc..

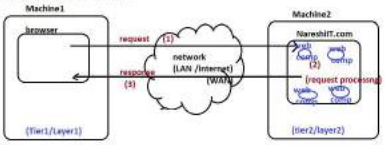
=> To devolope small scale JEE projects like java based small web sites for colleges , universities and etc.. take the support of webServer s/w like Tomcat
=>To devolope large scale JEE projects like java based e-commerce websites (like Flipkat) take the support of Aplication s/w like weblogic /Wildfly server..

=> Programmers need not devolop webServer /Application server s/ws.. becoz multiple software vendor companies already there in that market creating them releasing them to the market (50+ vendors)
=>Programmer must know how install and how to use webServer or Applicalton serve s/w to devleop software Apps (JEE aplications)

=>Using JEE module , JEE technologies based WebServer /Application server softwares , we can develop Enterprise Java Applications

any medium or large scale App with complexity is called an Enterprise App

Web applicalton (webSite)
=========================
=>Web applicalton is client - server App where Client is browser and server App is a s/w App interacting wtih each other in request-response model /way of comunication..
=>Browser gives requests , Webapplication (server s/w app) takes the requests ,process requests and delivers response back to browser..



=>Any reusable class/file is called comp

A logical or physical partiation on the Application representing certain logics is called tier/Layer.

eg:: NareshIT.com , gmail.com ,flipart.com, amazon.in , hyd.com and etc..

note:: we use Servlet,Jsp technologies of JEE module to devleop java based web applications

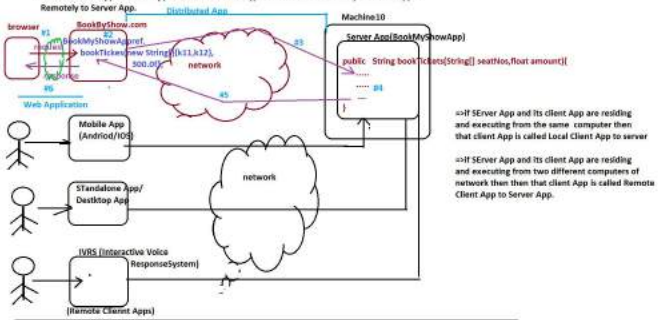
Distributed App/Remoting App
============================

JEE module and its technologies are useful to develop enterprise Apps

Distributed App
==========

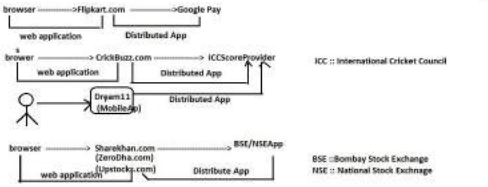
It is a client-server App where both client and server Apps are s/w Apps interacting with each other through method calls.

=>A Distributed App's Server App can have different types of clients either locally to Server App or Remotely to Server App.



=>If SErver App and its client App are residing and executing from the same computer then that client App is called Local Client App to server

=>If SErver App and its client App are residing and executing from two different computers of network then then that client App is called Remote Client App to Server App.



=> we use RMI (outed) ,EJB(outdated), CORBA (outdated),Webservices(latest) and etc.. technologies to develop Distribtued Aps.

RMI :: Remote Method Invocation
EJB ::Enterprise Java Beans
CORBA :: Common Object Request Broker Architcture

Java frameworks are not part JSE,JEE,JME modules , they are given third party companies/Vendors based on JSE,JEE module technologies to simplify JSE,JEE modlie Application devleopment.

=>JSF, Strutts, Spring MVC are frameworks that given based servlet,jsp technologies to simplify the java web application devleopment.

JSF:: java server Faces
spring MVC :: spring Model-View-Controller

Java learning is about
a) learning langauage (core java) (raw materials)
b) learning technology(adv java) (semi-finished products)
c) learnig fameworks(spring ,hiberanate,...) (fully finished products)

=> Axis, Apache CFX, Spring Rest and etc.. are the frameworks that are given based on webServices env.. to simplify the the development of Distributed Application Development

What is the difference b/w web application and Distributed App?

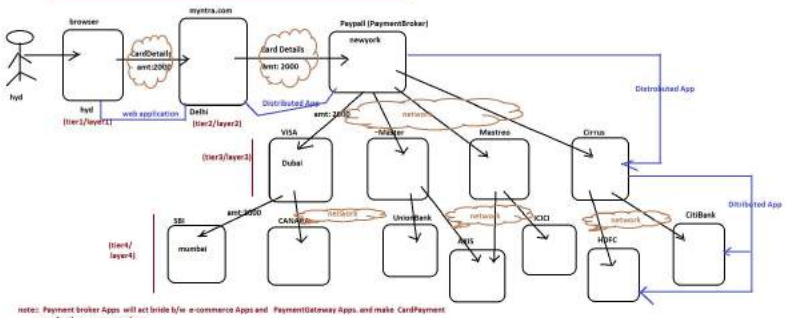
web application	Distributed App
a) It is client -server App where client is browser and server App is software App	a) It is Client -server App where both Client and SErver Apps are software Apps
b) It is Thin Client -Fat Server App	b) It is Fat Client -Fat Server App
c) Allow only Browsers as the Clients	c) Allow different types of Clients like WebSites, Mobile Apps, IVRS App,Desktop Apps and etc..
d) Communication takes place based on request-resppnse model	d) Communication takes place based on method calls/invocations.
e) To develop java web applications use Servlet,jsp (technologies) or struts, spring MVC, JSF (frameworks)	e) aTo develop java based distributed Apps , take the support of RMI,EJB,CORBA,WebSErvices (Jax-WS) (technologies) or spring rest, apache CFX , Axis (frameworks)

Adv.java
pre-requisite:: Core Java oops
duration is : 100+ sessions
daily 1:30 to 1.45 sessions
weekly 6/7 sessions
Course fee :: rs. 3500/-
Admin :: srikanth
contatct no : +91 9302 968 665 (only whatsup)

Backup videos ::
https://www.youtube.com/watch?v=2oTIHv-Gwfi&t=55s
class notes ::
https://www.facebook.com/groups/588093825162910/files
FB group name:: nataraajawaarena
email Id :: nataraajawaarena@gmail.com

An enterprise App can be a web application or distributed App or combination of both.. Generally enterprise Apps are n-tier Apps (multilayer Applicaitons)(

eg:: e-commerce with Card Paymemnt are examples for enterprise Apps (web applicaiton + Distributed App)



note:: Payment broker Apps will act bride b/w e-commerce Apps and PaymentGateway Apps. and make CardPayment easy for the e-commerce Apps..
eg:: paypal, payumoney , sefe pay , Razorpay and etc...
note:: Payment gatewayprovide world wide infrastructure to operate credit cards and debit cards.. But they will these cards customers in different countries through local banks..
eg: VISA, MASTER, CIRUSS , RuPay , MASTREO and etc..

JAva Fresher ::
==========
Core java + adv.java +oracle +spring/spring boot *
CRT (Aptitugda, reasoning and etc..)

Java Exprerienced (1+)
==========
Fresher topis + spring/spring boot (mandatory), hibernate Tools (15+) , Design patterns | (CRT not reuqired)

JAVa Experienced (2+/3+/4+)
==========
1+ topics + webSErices + microservices + Java Realtime Project | (CRT not required)

JAva Developers

JAva FullSTackDeveloper (for 2+/3+/4+/5+/.....)
==========
Java :: core java +adv.java + Spring/springBoot + MicroSrvices +webServices
DB :: oracle /mysql
testing :: selenium
UI technologies :: html,js, css, Bootstrap , jquery/angular /ReactJS
DevOPs Tools :: maven/gradle, Jenkins, Docker and etc...
Clould :: AWS (Amazon web services) /Azure

For FB notes::
https://www.facebook.com/groups/388095825162910/files
For Backup videos::
https://www.youtube.com/watch?v=eIu2eXxeCCU&list=PLVIOHNRLfiP9OITKTQuq3UWJNA_wOPIFr (3 to 4 videos)

FB group :: nataraajavaarena
gmail id :: nataraajavaarena@gmail.com

pre-requisite :: core java oops
duration :: 100+ sessions
daily 1.30 mins to 1.45 mins sessions
admin name :: srikanth sir
contact number :: +91 6302968665 (only whatsup)

Import modules of JAVA are
a) JSE module
b) JEE module
c) JME moudle

J2SE
J2EE
J2ME
(old names)

JME
====
mdk :: Micro/mobile development kit.

JME ::Java Micro Edition
=>This module installable s/w as mdk s/w
=> Latest version :: 2.x
=> This module is given to micro apps and mobile Apps.

=>The Apps that are small in size and big in performence and can be used attached/ Embeded in chips to control different type types devices like mechanical, eletrical, eletronic, automoible and etc.. devices supporting embeddes system programming and Artificial intelligence.

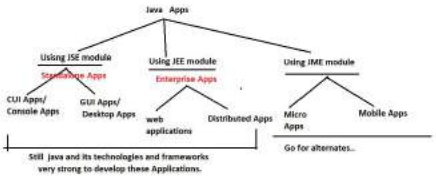
eg:: chips attached to the fully automated washing machine
chips attached to the mobile for GPS tracking
and etc..

=>Inside these chips we can place the code developed in JME module ...
The process of attaching chips to various of devices and making them to think and take decisions is called embeded system programming.. Which supports Artitificial Intelligence internally

=>Humans learn things by experience becoz we are having natural intelligence.. but Device do not have natural intelligence.. So we added Embedded System chips to devices to to make them to think and take decisions.. This is nothing but artificial Intelligence.

=>The Apps that run in mobile devices are called mobile Apps
eg:: mobile phoebook , mobile music player and etc..

JME module is outdated becoz that altenates that have come to the market ..
=>To develope Embedded System Apps supporting Artitificial Intelligence the market is
RTOS, VxWorks ,EmbeddedC , python and etc..
=>To develop mobile Apps , the market is using Adriod , IOS s/ws ..



What is the difference b/w Constructor and static block?

Ans) constructor is objecct level one time executing block .. where as
static block class level one time executing block.

=>When we create object for any java class.. the constructor executes and allows the programmer to initialize instance variables of object..

note:: If u want to establish connection with Db s/w for every object created for the java class , then place logic of establishing connection with DB s/w inside the constructor.

syntax :: <Modifier> <classname>(<params>){
... //initialization logic
...
}

=> When JVM loads the given java class then static block of the class will be executed automtically, we can static block to intialize static variables of the class... to write class level one execution logics..

syn::
static {
... //logic to intialize
... static variable
}

=>if I want to establish only once connection with Db s/w irrespective of no.of objects that are that created.. then we need to write such logics in the static block..

S/w setup
=========
=> any version jdk s/w (java14) (java8+ is good)
=>Editplus (any version) (2+ is good)

G:\Workspaces
|--->Advjava
|---->NTAJ415
|----->basics
|----->TestApp1.java

=>in one .java file, we can have multiple no.of classes... but there can be only one public class and that public class name and .java file name must match.

Class.forName("Test"); // makes jvm to load the given java class "Test" .. from HDD(hard disk drive) into JVM Memory of the RAM , If not available to load ,then we get exception called ClassNotFoundException

forName(-) is the static method of java.lang.Class
(class name it self Class)

=>Class.forName(-) makes jvm to load the given class java class but it does not any object for the loaded java class.

=>if java method throws checked exception (serious exception)then we should catch and hanle the exception using try/catch or we should declare the exception to be throwing using "throws"..

Example App
===========

```
//TestApp.java

class Test
{
    static{
        System.out.println("Test: static block");
    }
    public Test(){
    System.out.println("Test: 0-param constructor");
    }

}

class Demo
{
    static{
        System.out.println("Demo: static block");
    }
    public Demo(){
    System.out.println("Demo: 0-param constructor");
    }
}

public class TestApp
{
    static{
        System.out.println("TestApp: static block");
    }
    public static void main(String args[]) throws Exception{
        System.out.println("start of main(-) method");
         //create objs for Demo class
         Demo d1=new Demo();
         Demo d2=new Demo();
      Demo d3=new Demo();
    //Load Test class
       Class.forName("Test");
              Class.forName("Test");
              Class.forName("Test");
                 Class.forName("Demo");

        System.out.println("end of main(-) method");
    }//main
}//class
//cmd>javac TestApp.java (compilation)
//cmd>java TestApp (execution)
```

ouput::
```
TestApp: static block
start of main(-) method
Demo: static block
Demo: 0-param constructor
Demo: 0-param constructor
Demo: 0-param constructor
Test: static block
end of main(-) method
```

API (Application Programming Interface)

The word interface here in no way related java interfaces..
it is just used having english meaning "base" or "platform" or "foundation"

=> API base for the programmers to develop s/w Apps .. Every language, Technology , framework provides built-in APIs and allows the programmers to develop user-defined apis..

=> API in "C" language comes in the form of set of functions as header file (.h files)
=> API in "C++" language comes in the form of set of functions and classes as header files
=> API in "Java" language comes in the form of set of classes , interfaces, enum[2]and annotations as java packages

example java apis
=================
=> lang api (java.lang and sub packages)
=> utility api ((java.util and its sub packages)
=> IO streams api ((java.io and sub packages)
=> Reflection api ((java.lang.reflect and sub pckages)
=>JDBC API ((java.sql , javax.sql and their sub packages)
=>servlet api ((javax.servlet and its sub packages)
and etc..

=>APIs can be also called as Libraries and they will be developed and released to market in the form jar files..
((java archieve file ---> java based zip file)

=>All pre-defined apis of Jdk s/w are released as rt.jar file..
<java_home>\jre\lib folder (only upto java8)
[Java installation folder]

3 types of APIs
----
a) pre-defined APIs (will come along with language/framework/technology)
b) user-defined APIs (created by the programmers)
c) Thrid party APIs (Given by Third party companies)

example

=> All packages supplied by jdk (which are part of rt.jar) represents pre-defined apis | Given by Sun Ms/Oracle corp
eg: java.lang, java.util, java.io, java.net, java.awt and etc.. pkgs | (Party1)

=>Porgrammer creates packages having classes ,interfaces ,enums and annotations are called user-defined apis | (Given by Programmers) (party2)

=> Third party companies supplied java packages are called third party apis. | (Given by some other company (party3)
eg: apache DBCP, Hikari CP and etc..

=>In App developement ,we use APIs in the following order
a) pre-defined apis given by language or technology or framework
b) user-defined apis developed the programmers
c) Thrid pary apis given by Third party companies/vendors

jar (Java archieve)

=>It is java based zip file or rar file
=>java env.. jar files representing multiple things... they are

a) pre-defined or user-defined or third party apis (eg rt.jar represents pre-defined apis)
b) represents JDBC driver s/ws ..
(JDBC driver s/w is bride b/w java App and DB s/w converting java instructions to DB instructions and vice-versa)
c) represents java Apps/Projects that are packaged for release to Client organizations..
POLARIS realeases s/w project to citbank in the form jar file
(s/w company) (Client organization)
CTS releases the s/w project to AMEX in the form of jar file
(s/w company) (client org)
d) JAva based web sites will be hosted on to internet in the form of war file
war --> web application archieve (a flavour of jar file)
The jar file representing the web application/website in called war file

e) represents EJB comps (Distributed App)

f) Can be used to zip multiple files and folders into single file (like zip file)

and etc...

=> "jar" is built-in tool in jdk s/w like javac , java and etc.. which can be used to perform multiple operations on jar file like creating jar files, seeing the content jar file, extracting content from the jar file and etc..

in <java_home>\bin directory

Example operations using jar file

=>In jar files we can place any file and any file.. including .java files and .class file..

```
G:\WorkSpaces
    |--->advjava
        |----->ntaj415
            |--->JarDemo
                |--->a.txt
                |--->b.txt
```

To create jar file

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar  cvf  nit.jar  .
added manifest
adding: a.txt(in = 25) (out= 21)(deflated 16%)
adding: b.txt(in = 25) (out= 21)(deflated 16%)
```

. represents all files and folders of the current directory

'.' -->represents current directory
'..' --> parent directory..

c --> create archieve
f -->specify the archieve file name (jar file name)
v --> verbose mode (detailed message will come)

To see the content of jar file

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar tvf nit.jar
   0 Wed Apr 14 17:10:10 IST 2021 META-INF/
  68 Wed Apr 14 17:10:10 IST 2021 META-INF/MANIFEST.MF
  25 Wed Apr 14 17:07:38 IST 2021 a.txt
  25 Wed Apr 14 17:07:44 IST 2021 b.txt
```

t-->gives table fo content..

=>every jar file by default gets one manifet file that is manifest.mf in META-INF folder

info about current jar file..

manifest.mf

```
Manifest-Version: 1.0
Created-By: 1.8.0_91 (Oracle Corporation)
```

To add new file to existing jar file

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar  uvf  nit.jar  c.txt
adding: c.txt(in = 400) (out= 356)(deflated 11%)
```

u--> update archieve file (jar file) with new content

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar tvf nit.jar
   0 Wed Apr 14 17:10:10 IST 2021 META-INF/
  68 Wed Apr 14 17:10:10 IST 2021 META-INF/MANIFEST.MF
  25 Wed Apr 14 17:07:38 IST 2021 a.txt
  25 Wed Apr 14 17:07:44 IST 2021 b.txt
 400 Wed Apr 14 17:23:46 IST 2021 c.txt
```

To extract specific file from jar file

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar xvf nit.jar  a.txt
 inflated: a.txt
```

x-->extract from archieve file (jar file)

To extract all the files from jar file

```
G:\Worskpaces\advjava\NTAJ415\JarDemo>jar xvf nit.jar
  created: META-INF/
 inflated: META-INF/MANIFEST.MF
 inflated: a.txt
 inflated: b.txt
 inflated: c.txt
```

Q1) Why java is throwing NullPointerException .. though java is not supporting pointers?
Q2) What is the difference b/w System.out.println(-) and System.err.println(-) methods?

What is the difference b/w PATH and CLASSPATH?

PATH
=====
=> It is DOS/Window Command and can be used in any technology/framework/language
=> In order make exe files or bat files or cmd files of certain folder executing from any location of the computer, we need to add their address Location to PATH env.. variable.

problem::

G:\workspaces
|-->advjava
|----->NTAJ415
|------>basics
|--->PATHDemo
|--->run.bat

.bat /batch file is useful to combine set of commands into single command

run.bat
---
date
time
ver
dir

G:\Workspaces\advjava\NTAJ415\basics\PATHDemo>run.bat or run (success)
excutes all the 4 commands in sequence

G:\>run (error)
=> Bad command or file name
=> run is not a internal or external command or operable program
E:\Workspaces> run (error)
=> Bad command or file name
=> run is not a internal or external command or operable program
c:\>run (error)
=> Bad command or file name
=> run is not a internal or external command or operable program

run1.bat (another example of batch file)
======
cd E:\Workspaces\abc
G:
javac TestApp.java
java TestApp

g:\Workspaces
|--->abc
|---->TestApp.java
(class with main(-) method)

g:\workspaces\abc>run1.bat (or) run1

problem:: We must execute .bat file or .cmd file or .exe file by going to its orginal location.. i.e we can not execute them from any location of the computer

Solution1: (Bad)
=========
=>copy .bat file or .cmd file or .exe file to other locations of the computer where we want to execute..
=>It is bad solution becoz the copy process will waste the memory of the computer..

Solution2:: (Good)
==========
=>Add the locations of .bat files or .cmd files or .exe fiels to PATH env.. variables and we u can execute .bat file or .cmd file or .exe file from any location of the computer.

This PC /My Computer --->propeties --->advanced system settings ---> env.. variables-->
system variales ----> new
variable name :: PATH
value :: G:\Worskpaces\advjava\NTAJ415\PATHDemo;.
--->ok --->ok --->ok.

note:: java ,javac ,jar , javap and etc.. are the java tool given as .exe files in <JAVA_home>\bin directory.. So to make these tools (exe files) executable from any location computer we need to add <JAVA_HOME>\bin directory PATH env.. variable..

This PC --->properties ----> adv system settings ----> env .. variables ---> system properties --->
variablalbe name :PATH
value :: C:\Program Files\Java\jdk1.8.0_31\bin;G:\Worskpaces\advjava\NTAJ415\PATHDemo;.
-->ok --->ok --->ok

note:: PATH is not a JAva command .. So it can be used in both Java and non-Java env.. ..

CLASSPATH
==========
=> It is purely Java command .. and can be used only in Java env..
=> If our Java App uses both user-defined apis/libraries or third paty apis /libraries then we need to add those apis related directory locations or jar files to CLASSPATH env.. variable in order to make Java tools like javac, java and etc.. recognizing and using user-defined, thrid paty apis/libraries.

Problem::
=========

User-defined API/Library (Developed by Programmer1)
(1 pkg with 1 class)

G:\workspaces
|--->advjava
|--->ntaj415
|--->CPATHDemo
|---->WishMessageGenerator.java
|---->com.nt.basics
|--------->WishMessageGenerator.class

WishMessageGenerator.java

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

G:\Workspaces\advjava\ntaj415\CPATHDemo>javac -d . WishMessageGenerator.java

makes javac compiler
to create specified package (com.nt.basics)
in the specified current directory(.) and keeps
the generated .class file (WishMessageGenerator.class) in that
package (com.nt.basics)

Main App (Developed by Programmer2)

E:\nit
|-->MainApp.java

MainApp.java

```
import com.nt.basics.WishMessageGenerator;

public class MainApp{
  p s v main(String args[]){
     WishMessageGenerator generator =new WishMessageGenerator();
      S.o.println("wish message is :"+generator.sayHello("raja"));
  }//main
}//class
```

E:/nit>javac MainApp.java (errors)

Can not find symbols "com.nt.basics", "WishMessageGenerator","sayHello(-)".....

Problem:: The Javac of MainApp.java file compilation unable find the package "com.nt.basics" and its class "WishMessageGenerator" becoz they are not part of jdk apis and they are not even there in the current directory(E:/NIT).. So the compile error will be generated.

Problem::
=========

User-defined API/Library (Developed by Programmer1)
(1 pkg with 1 class)

G:\workspaces
|---->advjava
|---->ntaj415
|---->CPATHDemo
|---->WishMessageGenerator.java
|---->com.nt.basics
|---->WishMessageGenerator.class

WishMessageGenerator.java

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

G:\Workspaces\advjava\ntaj415\CPATHDemo>javac -d . WishMessageGenerator.java

makes javac compiller
to create specified package (com.nt.basics)
in the specified current directory(.) and keeps
the generated .class file (WishMessageGenerator.class) in that
package (com.nt.basics)

Main App (Developed by Programmer2)

E:\nit
|---->MainApp.java

MainApp.java

```
import  com.nt.basics.WishMessageGenerator;

public class MainApp{
   p  s v main(String args[]){
       WishMessageGenerator generator =new WishMessageGenerator();
       S.o.println("wish message is :"+generator.sayHello("raja"));
   }//main
}//class
```

E:/nit>javac MainApp.java (errors)
Can not find symbols "com.nt.basics", "WishMessageGenerator","sayHello(-)"....

Problem:: The Javac of MainApp.java file compilation unable find the package "com.nt.basics" and its class "WishMessageGenerator" becoz they are not part of jdk apis and they are not even there in the current directory(E:/NIT).. So the compile error will be generated.

Solution:: Add the location of "com.nt.basics" pkg (nothing but G:\workspaces\advjava\ntaj415\CPATHDemo folder) to CLASSPATH env.. variable..

This PC --->properties ----> advanced system settings --->env.. variables ---->
system variables ------>
variable name :: CLASSPATH
value :: G:\workspaces\advjava\ntaj415\CPATHDemo; .
-->ok --->ok --->ok

E:/nit>javac MainApp.java (success)
E:/nit>java MainApp (success)

The modifications done in env.. variables will not be reflected to already opended old command prompts.. So prefer opening new command prompts..

Conclusion:: With respect java ..
=> To use java in computer we need to add <JAVA_HOME>\bin directory to PATH env.. variable
=> To make Java Apps working with user-defined apis and third party apis .. work with CLASSPATH variable by adding user-defined /third api directories /jar files to CLASSPATH env.. variable.

Creating jar file representing the above user-defined api

G:\Workspaces\advjava\NTAJ415\CPATHDemo>jar cvf wishlib.jar .
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/nt/(in = 0) (out= 0)(stored 0%)
adding: com/nt/basics/(in = 0) (out= 0)(stored 0%)
adding: com/nt/basics/WishMessageGenerator.class(in = 503) (out= 318)(deflated 36%)
adding: WishMessageGenerator.java(in = 183) (out= 135)(deflated 26%)

=>if the user-defined api or third party api is there in the form of jar file then must add that jar file (including location and filename) to CLASSPATH env.. variable

This PC --->properties ----> advanced system settings --->env.. variables ---->
system variables ------>
variable name :: CLASSPATH
value :: G:\workspaces\advjava\ntaj415\CPATHDemo\wishLib.jar;.
-->ok --->ok --->ok

E:/nit>javac MainApp.java (success)
E:/nit>java MainApp (success)

Important points on Env.. variables

a) Environment variables are not case-sensitive i.e they can be there in upper case or in lower case or in mixed case
b) user env.. variables are specific to currently logged in windows user. where as system env.. variables are visible for multiple windows users of a computer
c) The modifications and additions done to env.. variables will be not reflected to already opened old command prompts. So we must open new command prompts..
d) Values set env.. variables from cmd prompt will remain temporarly .. where as the values set env.. variables of This PC/my computer will remain perminently..
e) multipe values added to the env.. variables must be seperated with ";" symbol
f) It is recomanded to add new values to the env.. variables at the begining of existing values
g) "." represetns current directory , ".." represents parent directoy when they are added to env.. variables as values.

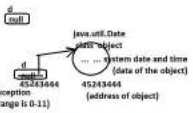
and etc..

When java is not supporting pointers ,why it is throwing NullPointerException?

Ans)

problem code::
```
java.util.Date d=null;
 int month=d.getMonth();  //Throws NullPointerException
```

solution code::
```
java.util.Date d=null;
   d=new java.util.Date();
   int monthnd.getMonth();  //Does not Throw NullPointerException
                  |---> current month number will come  3  (range is 0-11)
```



note:: The word pointer in NullPointerException does not presenting c,c++ pointers.. it is used having english meaning indicating that the method is invoked on the reference variable that pointing to NULL value.

What is the difference b/w System.out.println(-) and System.err.println(-)?

Ans) S.o.p(-) supports output redirection .. where S.e.p(-) does not support output direction..

```
//App1.java
public class App1{
   public static void main(String  args[]){
          System.out.println("hello1");
          System.err.println("hello2");
   }//main
}//class
```

cmd>javac App1.java
cmd> java App1 > xyz.txt
hello2 (S.e.p generated message)

xyz.txt
hello1 (S.o.p generated message)

Redirects the output to the given xyz.txt file from the console

cmd>java App1 >>xyz.txt
redirecting with append activity..

Q) How many ways are to read inputs from enduser through our Java App?
Q) How find out the java version of give .class?
Q) Can we developa java App with out using any kind of user-defind class?
Q) can we develop java App with out using main(-) method?

JDBC
=====

old days :: JDBC means Java DB Connectivity
nowadays :: JDBC is a trademark
No abbrivation/ No acronym

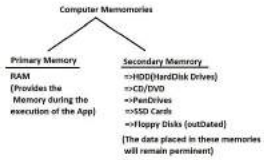
=>It is given a JSE module technology having APIS (packages with classes ,interfaces ,enums and annotations) to interact with DB s/w and to perform persistence operations on DB s/w.
(oracle , mysql, postgresql and etc..)

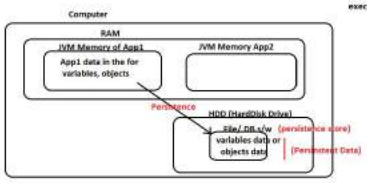
persistence
==========

=> It is the process stroing and managing data for long time

=> The data stored in variables, objects of the App allocate memory in the JVM memory of the RAM.. Since RAM provides memory only during the App execution.. So the data in variables/objects will be vanished at the end of execution i.e the data of the App can not be used across the multiple executions of same App or different Apps.

=>App data means the inputs given to App and the outputs/results generated by the App.





PErsistence store
=================

=>The place/store where Data can be saved and managed for long time is called Persistence store

eg: files , Db s/w and etc..

Persistence stores generally use secondary devices like HDD, CD/DVD and etc.. to store and managed data for long time on Perminent basis.

Persistent Data
===============

=>The data of PErsistence store that is saved and can be reused is called persistent data..

eg: file's content , Db s/w's db tables and their records

Persistence operations
======================

=> The insert,update,delete and read opertaions performed on the persistent data of Persistence store are called Persistence operations..

These operations are also called as CURD/CRUD/SCUD Operations

CURD/CRUD
=========
C---> create (insert)
U ---> Update (modify)
R ---> Read (select)
D --->Delete (remove)

SCUD
=====
S --->Select
C --->create
U --->Update
D--->delete

=> All service sector software project like Banking Apps, Insurance Apps, Ticket Reservation Apps, Food delivery apps and etc.. definetly deal CURD Operations while offererring services customers.

Persistence logic
=================

=>The logic that is developed to perform Persistence operations on persistent data is called Persistence logic

eg:: IO stream code , jdbc code , hibernate code , spring jdbc code , spring ORM code, spring data JPA code..

Course :spring Basics, Spring Boot, MicroServices

Persistence technology/framework
================================

=>The technology/framework using which we can develop Persistence logics is called Persistence technology /framework

Java Learning is all about
language (core java)
technologies (adv.java)
frameworks (spring, hiberate, webservices and etc..)

=>IO Stream (part of java language)
=>JDBC (Persistence Technology)
=>Hibernate ,spring JDBC, spring ORM, spring dataJPA (Persistence frameworks)

ORM :: Object relational mapping
JPA :: Java Persistence API
(Application Programming Interface)

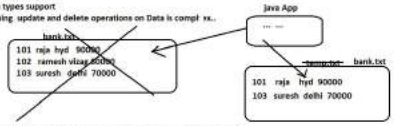
=>Persistence technology/framework is given for developing Persistence logics to perform persistence operations on the Persistent data of Persistnece stores.

Java App ---(IO streams (java.io pkg))---> Files (Flat files)
=>The files that are not specific to any Operations OS or any Software
(Mostly these text files)

Java App ---(JDBC +SQL (ORM))---> DB s/w (oracle ,mysql, postgreSQL and etc..)
Object relational Mapping like Hibernate

Limitations with flles as Persistence stores
==

a) Can not store huge amount of data
b) No security
c) No SQL support
d) No Constraints support
e) No Data types support
f) Performing update and delete operations on Data is compl ex..



g) Performing select operations with multiple conditions is very complex.
h) One file data can not kept in relationship with other file data. (No Foreign key support)
i) Data merging across the multiple files is very difficult
j) Data comparision among the multiple files is very complex.
and etc..

=>To overcome these problems use Db s/w as the Persinstence store.

In realtime java projects, where did u use files as the persistence store and where did u use Db s/w as the Persistence store?

In realtime java projects, where did u use files as the persistence store and where did u use Db s/w as the Persistence store?

=> For small scale Apps that are generating little amounts of data to persist then prefer using files as the persistence stores..

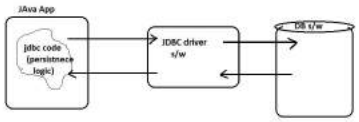
eg:: mobile Apps(phonebooks ,music player and etc..), mobile games, standalone Apps like Desktop games and etc..

=> For medium and large scale Apps that are generating huge amounts of data to persist use DB s/ws as Persistence stores

eg:: Banking Apps , e-commerice Apps(flipkat, amazon and etc..) , Insurance Apps, web applications, enterprise Apps, distributeted Apps and etc..

Most of Java ,jee and java framework projects use Oracle or mysql or postgreSQL DB s/w as persistnece store...

For Java App to DB s/w interaction , we need JDBC driver s/w as bridge s/w to convert Java calls (java instructions) to DB calls (DB instructions) and vice-versa .



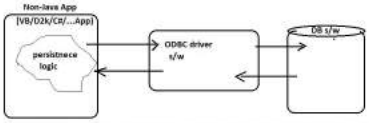
=>Device drivers are the bridge b/w Operating system s/w and external/internal devices of the Computers they convert OS calls(instructions) to Device calls (instrctions) and vice-versa.

eg:: printer drivers/printer device drivers , Sound drivers , VGA drivers and etc..

Generally , we need seperate JDBC Driver s/w for interacting with every DB s/w from Java App

=>Java App uses JDBC driver s/w for oracle to interact with Oracle Db s/w
=>Java App uses JDBC driver s/w for mysql to interact with MYSQL Db s/w
=>Java App uses JDBC driver s/w for postgreSQL to interact with PostgreSQL Db s/w and etc...

Non-JAva Apps like VB Apps, d2kApps, cRApps use ODBC driver s/ws to interact with DB s/w..



ODBC (Open DB connectivity) driver s/w act as bridge b/w Non-Java App and Db s/w to convert Non-Java App calls to DB s/w calls and vice-versa..

For Every Db s/w , we use seperate ODBC driver s/w to interact from Non-Java App

Non-Java App uses ODBC driver s/w for oracle to interact with Oracle Db s/w
Non-Java App uses ODBC driver s/w for mysql to interact with MYSQL Db s/w
Non-Java App uses ODBC driver s/w for postgreSQL to interact with PostgreSQL Db s/w and etc..

=> ODBC driver s/ws are given to market before the arrival JDBC driver s/ws
ODBC driver s/ws are given around 1990 year
JDBC driver s/ws are given around 1996 year.

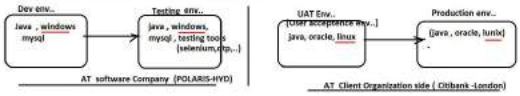
Q) When all other Non-Java languages/technologies based Apps are using ODBC Drivers to interact with Db s/ws.. why Java Apps need seperate/exclusive JDBC driver s/ws to interact with Db s/w.

Ans) ODBC driver s/w are developed in "C" language having support of pointers by group of companies called ODBC Group (32+ companies are there in the group including Microsoft, IBM and etc..)

=>initially Java people also thought of using ODBC driver s/ws to make Java Apps interacting with Db s/w.. but they failed to use becoze of the following reasons..

a) ODBC driver are devloped in "C" language taking the support of pointers but JAVA does not support pointers
b) Converting Pointer notation logic of "C" language to Non-Pointer notation logic of Java is very complex and kills the performence of DB s/w connectivity
c) "C" language code Platform/OS Dependent, So using ODBC driver s/w along with Java App makes that Java App indirectly Platform dependent.

Various env.. in Java Project development and release



In UAT env.. the recieved/released project from software service company will be installed in Client org machine (CitiBank- london) , some dummy users will be created in the project to make the technical team of Client organization to test the project for multiple days.. If UAT results are positive the project goes to production (Use for customers and employees of Client organization) otherwise project will be given Development team to fix the bugs and issues.

Why not windows can be used in Production env..?

Ans) In one windows OS based computer, we can keep max of 7 micro processors to improve computer speed.. but that is not sufficient for the production env.. projects which will be used lakhs of users at time ..

In One Linux OS based computer , we can keep 64 no.of parellel MicroProcessors to improve the computer speed ... which definetly suitable for Proudction env.. projects.. So the Production env... always takes place in Linunx platform/OS.

=>we can get ODBC driver s/ws from the following software Vendor companies
(The companies who creates softwares)
a) ODBC Group (32+ companies togather is called ODBC group) (Best)
b) DB s/w Vendors (The vendor Companies who creates and releases
DB s/ws to the market)
c) Third party Vendors (like Devx , Apache and etc..)

=> We can get JDBC driver s/ws from the following vendor companies
a) Sun Ms (Only in the intial days java/JDBC release)
b) DB s/w vendors (Best)
c) Third party Vendor (like apache, Devx and etc..)

What is the difference b/w Programming language and software Technology?

Programming language
==========

=> Programming language/software language is directly installable s/w as software raw materials providing basic features to develop the s/w Apps.
=> Programming language define syntaxes and semantics of the programming by supplying compilers and interpeters..

syntax rules :: rules of the programming
eg: every instruction should end with ";" symbol, every { must have } and etc..

semantics : structure of programming
eg:: order of content in programming
in Java Prg/app :: package decls, packge imports , class defination, variables decl/ constructor def/ method decl or defination

=>Pogramming /software languages base to create s/w technologies , frameworks , Tools , DB s/w , operating systems and etc..

=>andriod is developed using c++ and java
=>IOS is developed using Pro "C" language
=>Linux/Unitx Operating systems are created using "C" languages
=> Java Technologies like JDBC,Servlet,Jsp and etc.. and Java frameworks like spring,hibernate and etc.. created using java language
and etc..

eg:: c ,c++,java, java script , c#, ada and etc..

JVM based languages are
java, groovy, go, scala, kotlin and etc..

All JVM based languages generate .class file after compilation and they executed using same JVM.

S/w technology
==========

=> S/w technology is a software specification that gives set of rules and guidelines in the form APIs to create implementation softwares by using one or another programming language.
=>S/w technology is not installable , but the s/w technology based implementation software is installable or arrangable.. Working with this implementation s/w is nothing but working with software technology.

eg:: JDBC, Servlet, jsp ,EJB and etc...

=>JDBC is a JSE module supplied Java technology that gives set of rules and guidlines in the form of JDBC API to create JDBC driver s/ws by using java Programming language.
=>JDBC technology is not installable .. But the JDBC technology based JDBC driver s/ws are installable or arrangable, So working with JDBC driver s/ws is nothing but working with JDBC Technology.

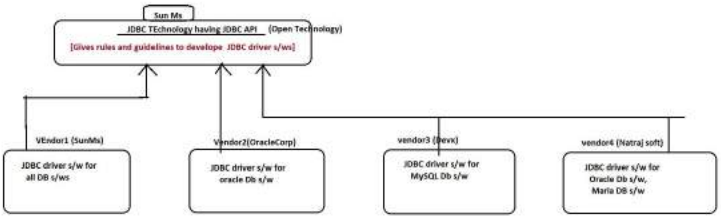
In JAva Technology API (packages) (upto java7)
=> interfaces represent rules
=> concrete classes represent guidlines
=> abstract classes represent rules and guidelines
In Java Technology API(packages) (from Java 8)
=> interfaces represent both rules and guidlines (Java 8 onwards interfaces can have default,static methods with definitations)
=> concrete classes represent guidlines
=> abstract classes represent both rules and guidelines

There are two types of S/w technologies

a)Open Technologies
=>Here the rules and guidelines of s/w technology are open to all the s/w vendors to create implementing s/ws.. (It like scientist keeping formuale in the open/public youtube channel and allowing all the vendor companies creating products based on the formulae)
eg:: All JSE (JDBC,JNDI,RMI and etc..) ,JEE(Servlet,jsp,EJB,...and etc..) TEchnologies are open technologies

b) Proprietary Technologies
=>Here the Vendor who has created s/w Technology is only allowed to create the implementation softwares.. i.e other s/w vendors can not create implementation softwares by using the techology rules and guidelines

eg:: All Microsoft technologies like asp.net, vb.net and etc..



How Open source and Open technology vendor companies are surviving financially?

Ans) They do not charge for main softwares but they charge for supplimentary s/ws and for services

=>Java is Open soruce i.e jdk s/w is free , more ever we get entire soruce code along with JDK s/w installation (src.zip) but it generates revenue/money for Oracle corp in the following areas
=> By offering java certification exams
=> By offering training for java certification exams
=> By giving corporate training on new technologies/frameworks of java for working employees..
=> By publishing books
=> By giving software services to the s/w companies if the project strucked in the middle of development
and etc..

=>Since all Vendor companies giving their JDBC driver s/ws of different DB s/ws based on the common rules and guidelines of JDBC Technology /API ,So the way work with all the JDBC driver s/ws to interact with DB s/ws is very similar i.e if we know how to use one JDBC driver s/w given by one Vendor we can use that knowledge to work with other JDBC driver s/ws given by other vendors..

[It is like learning Four wheer driver using maruthi 800 car and becoming ready to driver any model car given by car manifacturing vendor becoz all Car manifacturing vendors are manifacturing their cars of diferent models based on common rules and rules and guidelines car specificication]

What is JDBC ?

=> It is a J2E module supplied Open Java Technology from Sun Ms providing rules and guidelines in the form of JDBC API to create JDBC driver s/ws for different DB s/ws.

what is ODBC?

=> It is Open Technology given by ODBC group (32 + companies) having rules and guidliens to create ODBC driver s/ws for diferent Db s/ws..

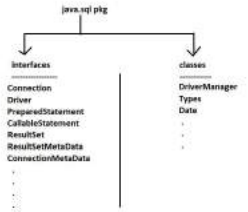
JDBC API Latest version : 4. x

JDBC API given JDBC technology is having the following packages

java.sql JDBC Core pkg i.e base pkg for other pkgs. 4.0 (four dot oh)

javax.sql

javax.sql.rowset in javax .. the "x" represents "extended pkg"



=>Implementing various interfaces of JDBC API is not the responsibility of Programmers .. it is the responsibility of the Vendor companies who are creating JDBC dirver s/ws...

=> Creating JDBC driver s/ws using JDBC API based rules and guidlines is the responsibilty of S/w Vendor companies but taking them and uses them in Java Apps to interact with Db s/ws is the responsibilty of Programmers..

How s/w Vendor companies develop JDBC driver s/ws for different Db s/ws?

=> Every Vendor company takes jdbc api pkg interfaces and provides classes implementing those interfaces of jdbc api .. In those classes logics will be kept to locate certain Db s/w , to establish connection with Db s/w , to send and execute SQL queries to that Db s/w and to gather SQL results from that Db s/w

At the end all those implemnting classes will be packed into jar file and that files will be released to the market as JDBC driver s/w of certain Db s/w.

Vendor1 (oracle corp) -->creates jdbc driver s/w for oracle --> gives ojdbc6/7/8/...jar file
(contains implementing classes having logics to interact with oracle Db s/w)

Vendor2 (Devx) -->creates jdbc driver s/w for mysql ---> gives mysql-connector-java-<ver>.jar file
(contains implementing classes having logics to interact with mysql Db s/w)

Vendor3 (enterprise DB) -> create jdbc driver s/w for postgresql --->gives postgresql-jdbc-<version>.jar
(contains implementing classes having logics to interact with postgreSQL Db s/w)

How to use JDBC driver s/w in our Java Apps?

a) download jar file representing jdbc driver s/w from internet like ojdbc6/7/8/....jar

b) add that jar file to the classpath (add ojdbc6/7/8/....jar file to classpath)

c) Activate JDBC driver s/w in our JAva App using java code
(will be discussed in future)

d) Use the activated JDBC driver s/w and JDBC API to develop
JDBC Persistence logic [interacting Db s/w and performing CURD Operations]

=>Every JDBC driver s/w is collection of java calsses implementing various interfaces of JDBC API .. But every JDBC driver s/w is indentified with JDBC driver class name.. Either by loading that jdbc driver class or by creating object for that JDBC driver class we can activate JDBC driver s/w in Java App.

=>The java class that implements java.sql.Driver(I) directly or indectly is called JDBC driver class name.. we can activate jdbc driver s/w in our Java Apps either by loading this jdbc driver class or by ceating object for this Jdbc driver class.

=>Oracle corp supplied JDBC driver s/w for oracle is comming in the form of ojdbc6/7/8...jar file.. In this JDBC driver s/w that jdbc driver class name is "oracle.jdbc.driver.OracleDriver"

pkg name JDBC driver class implementing java.sql.Driver(I)

App1.java (make sure that ojdbc6/7/8...jar file is added to CLASSPATH env...)

we are activing JDBC driver s/w for oracle here..

```
Class.forName("oracle.jdbc.driver.OracleDriver");
        (or)
oracle.jdbc.driver.OracleDriver driver=new oracle.jdbc.driver.OracleDriver();
```

Q) What is the difference b/w jdbc driver s/w , jdbc driver class and jdbc driver interface ?

Ans) =>JDBC driver driver s/w is bridge b/w Java App and DB s/w coverting java calls to DB calls and vice-versa
=> Every JDBC driver s/w can be identified and can be activated using JDBC driver class name
=> The java class that implements java.sql.Driver(I) (jdbc driver interface) is called JDBC driver class name.

=>Every Jdbc driver s/w will released in the form jar file having unique jdbc driver class name..But All the jdbc driver classes given by different JDBC driver s/w will implement the common java.sql.Driver(I)..

jdbc driver s/w given by oracle corp
jar file ::ojdbc<n>.jar
jdbc driver class name:: oracle.jdbc.driver.OracleDriver

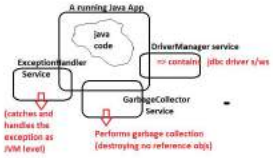
java.sql.Driver(I)
↑ implements
oracle.jdbc.driver.OracleDriver(c)

Devx supplied jdbc driver s/w for mysql
jar file name:: mysql-conenctor-java-<ver>.jar
jdbc driver class name :: com.mysql.cj.jdbc.Driver (pkg name / class name)

java.sql.Driver (I)
↑ implements..
com.mysql.cj.jdbc.Driver(c)

DriverManager service

=> Every Java App contains an invisible built-in service called DriverManager service having capability to manage set of JDBC driver s/ws..
=> Java App can use JDBC driver s/w only after registering with DriverManager Service directly or indirectly
=> java.sql.DriverManager class reprsents DriverManager serivce i.e using this class we can access drivermanager service to perform various activities in it like registering jdbc driver s/w , deresistering jdbc driver s/w , esablishing connection with Db s/w using one of the registered jdbc driver s/w and etc..
=> Java App is having multiple built-in and invisible services like driver manager service , garbage collector, exception handler service and etc..
=> Using System.gc() or Runtime.gc() we can invoke garbage Collector service..



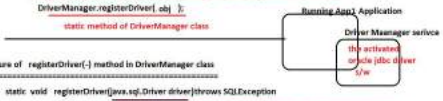
=>To place or register JDBC driver s/w with DriverManager service we need to place JDBC driver class obj with DriverManager service by Calling DriverManager.registerDriver(-) method (static method of DriverManager class)

Code to register oracle corp supplied JDBC driver s/w with DriverManager service through our Java App

In App1.java (make sure that ojdbc<n>.jar file is added to classpath)

```
//create jdbc driver class obj
oracle.jdbc.driver.OracleDriver obj =new oracle.jdbc.driver.OracleDriver();
// register jdbc driver s/w with DriverManager by keeping jdbc driver class obj
DriverManager.registerDriver( obj );
```

static method of DriverManager class



signature of registerDriver(-) method in DriverManager class

```
public static void registerDrivar(java.sql.Driver driver)throws SQLException
```

=>While calling this method , we can pass any jdbc driver s/w supplied jdbc driver class obj as the argument value becoz all the jdbc driver classes are the implementing classes of java.sql.Driver(I) ..

3 imp points on java method parameter types

a) if the parameter type is an interface, then we must pass one implementation calss obj of that interface as the argument value. while calling the method (refer above)

```
public void m1( X x)
m1(new A());
m1(new B());   | valid
m1(new X());   //invalid
```

A implements X (I)
B implement X (I)

b) if the parmeter type is an abstract class, then we must pass one sub class object of that abstract class as the argument value while calling the method.

```
public void m1(Y y)
m1(new A1());
m1(new B1());   | valid
m1(new Y());  //invalid
```

A1 exxtends Y(AC)
B1 extends Y (Ac)

note:: in JAva we can't create objects for interfaces , abstract classes..

c) if the parameter type is concrete class(normal class), then we must pass either that concrete class obj or one of its sub class class obj argument value while calling the method.

```
public void m1(Z z)
m1(new A2());
m1(new B2());   //valid
m1(new Z());
```

A2 extends Z(C)
B2 extends Z(C)

STandard steps to develop Java App with JDBC code (JDBC App)

step1) register JDBC driver s/w with DriverManager service to activate JDBC driver s/w
[Keep JBC driver class object in DrvierManager service]

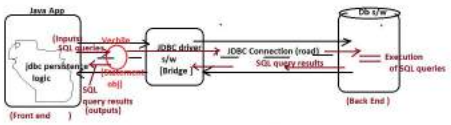
step2) Create Connection between Java App and DB s/w through JDBC driver s/w (bridge)
[It is creating/establishing road b/w Java App and Db s/w]

step3) create JDBC Statement object
[It is vechile b/w Java App and Db s/w.. it sends SQL querles to DB s/w from Java App
as inputs , executes them in Db s/w and gets their results back to Java App as SQL quey results]

step4) Send and execute SQL querles to DB s/w from Java App as inputs and executes then in Db s/w
using STatement object.

step5) Using Statement obj gather SQL Query results from DB s/w and process the results

step6) Close connection with DB s/w..

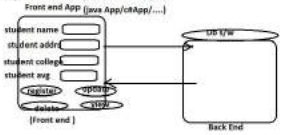


=>In any App what enduser can see and can operate is called front end (eg:: java App, .net App, php app and etc..)
=> in any App what enduser can not see and can not operate directly but it is part of the Project/App and responds for the front end instructions like CURD operations (Insert,update,delete and select oprations)
is called Back End (eg: DB s/w, files and etc..)

we can perform CURD operations on in Db s/w from SQL prompt of Db s/w like SQL Developers.. then why should we develop front end Apps in any technology/lanaguage/framework doing the same?

Ans) 1) Endusers are non-technical people and they can not write the SQL queries directly in DB s/w to complete the task.. so they need either CUI or GUI frontend to operform various operations on Db s/w..

=>The person who operates our app is called enduser .. He is non-technical person like bank employee,bank customer , mobile app users , website visitors and etc...



2) If enduser gets direct access to Db s/w to manipulate Data .. then he will be having unrestricted access to manipulate Db s/w data.. and in that process data of Db s/w can be misused ....

To overcome these Apps allow endusers to operate Db s/w data only through fronend End Apps where enusers feels easy to operate the Apps and he gets restricted access to manipulate the Db s/w data...

=>jdbc code in Java App makes standalone App talking with Db s/w
=>jdbc code in Servlet,jsp comps makes java web Apps talking Db with s/w
=>jdbc code in webservices comps/EJB comps makes java Dstributed Apps talking Db with s/w

[JDBC , on its own can not develop any kind of complete Java App.. But we can add JDBC code in all kinds of Java Apps like standalone,web , distributed Apps to make them talking to Db s/w..

The s/w vendor companies can use 4+1 mechanisms/methodologies/architectures to create JDBC Driver s/w for different Db s/ws by using the rules and guidelines of JDBC technology that are given in form of JDBC API.

They are

1)Type1 JDBC driver (JDBC-ODBC Bridge Driver)

(outdated from java8 i.e after java7 no vendor is supplying type1 machasim based JDBC driver s/w.. Upto Java7 only Sun Ms itself created type1 mechanism based JDBC driver s/w and given along with JDK s/w installation..Since no one is using this jdbc driver they stopped giving type1 JDBC driver s/w from java8 onwords)

2) Type2 JDBC driver (Native API/Partly Java Driver)

=>There are multiple s/w vendors giving Type2 mechanism based JDBC driver s/ws for diffferent DB s/ws..

=>Native API represents native methods i.e methods are declared in Java and implemented outside of java (like c,c++ and etc.. kanguages) and can be called from java apps

eg:: System.out.println(-)

note:: This JDBC driver s/w is partially developed in Java becoz of native API/native methods utilization

3) Type3 JDBC Driver (NwtProtocol /all Java Driver)

=>There are multiple s/w vendors giving Type3 mechanism based JDBC driver s/ws for diffferent DB s/ws..

=> This JDBC Driver is totally developed in Java and uses networking protocols like TCP/IP for locating different Db s/ws and other server s/ws..

4) Type4 JDBC driver (NAtive Protocol /all Java Driver)

=>There are multiple s/w vendors giving Type4 mechanism based JDBC driver s/ws for different DB s/ws..

=> These JDBC driver s/w are completed developed in Java and uses each DB s/w sepecific native prototols like "jdbc:oracle", "jdbc:mysql" and for locating and intaacting with DB s/w..

5) TypeS JDBC driver (No tecnnical name becoz it is not officially recognized by SunMs/Oracle corp)

=>There are multiple s/w vendors giving Type5 mechanism based JDBC driver s/ws for diffferent DB s/ws..

protocol
=========
=>set of rules followed by two parties in order to communicate with each other.
=> Two types protocols
a)network protocols::
=> Gives bunch of rules to get interaction among physical computers of a network
eg: TCP/IP , UDP and etc.. TCP/IP :: TranmissioeControlProtocol/Internet Protocol
b) application protocols:: UDP::User Datagram protocol
=>Gives bunch of rules to get interaction b/w two software services

eg:: http , jdbc:oracle , jdbc:mysql , smtp and etc..

http:: HyperTextTransferProtocol
(To get interaction b/w browser s/w and website)
smtp :: simple mail transfter protocol
(To get interaction b/w App and mail server (where email messages will stored)

note:: Application protocols run on top of network protocols

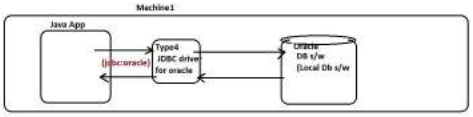
TCP/IP :: TCP is main protocol
IP is sub protocol
jdbc :oracle :: "jdbc" is main protocol
"oracle" is the subprotocl

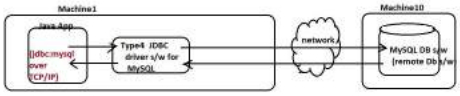
Type4 JDBC Driver Architecture
==========

=> This JDBC driver s/w is given to locate and interact with local or remote Db s/w directly by using Db s/w specific native protocols like jdbc:mysql, jdbc:oracle and etc...

=>This JDBC driver s/w is totally developed in Java ,So it is platform independent..

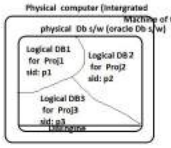
=> For every Db s/w, we need seperrate Type4 JDBC driver s/w i.e for oracle we need seperate Type4 jdbc driver s/w for oracle, for mysql we need seperate type4 jdbc driver s/w for mysql and etc..





Physical DB s/w vs Logical DB

=> Logical DB is a logical partition of physical Db s/w .. Every Logical partition is called one Logical DB and it will be created on 1 per s/w Project basis.. Every Logical DB contains db tables, PL/SQL procedures,functions and sequences related to each DB s/w.



In oracle Db s/w (physical Db s/w) , every Logical DB is identified with its SID (service ID) (nothing but technical name for Logical DB name)

=> If the 10 projects of a s/w company are using same oracle physical Db s/w.. then it would be installed only for 1 time in a common machine and multiple Logical DBs (10) will be created for mutiple projects (10) on 1 per each Db s/w basis

=>The installation of oracle Db s/w contains 1 default Logical DB and its default SID (If not changed during installation)is XE (for Expression Edition oracle (oracle free s/w))
ORCL (for Enterprise Edition of oracle (oracle paid s/w))

To find out SID of default Logical DB in oracle Db s/w after installation

```
SQL> connect system/manager
Connected.
SQL>
SQL>
SQL> select * from global_name;

GLOBAL_NAME
--------------------------------------------------------------------------------
XE  ----->(SID)
```

Port Number Vs Socket Number

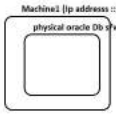
=> Every Physical computer contains Hardware ports (physical) to connect hardware devices to computer
=> Operating System installed in the computer will manage all the s/w services on different software ports (logical ports). Every s/w port is identified with its s/w port number..
=>In Window OS total 65,535 software ports are available.. In that starting 1024 (1 to 1024) are reseved for OS services.. and remanaing s/w ports (1025 to 65,535) are given for extenally installed software services..

external s/w service	default port number
Tomcat	8080
oracle	1521
mysql	3306
postgreSQL	5432

C:\WINDOWS\system32>netstat -a (open cmd as admin)

Socket Number of s/w service :: <IPAddresss>/<Hostname of computer> +< s/w service port number>

=>Socket number is extension of software portnumber..



=>oracle s/w port number :: 1521
=>oracle s/w socket number :: 176.99.66.55+1521
(Ip address) (s/w port)

=>oracle s/w socket number :: Machine1+ 1521
host name

note:: computer in the network is identified with its IP addresss or host name..

=>Oracle corp supplies both Type2 and Type4 mechanism based JDBC drvier s/ws for oracle Db s/w
along with oracle Db s/w installation,
=>Oracle corp supllied Type4 JDBC driver s/w for oracle is called as "oracle thin driver" and simliarly
Type2 jdbc jdbc driver s/w for oracle is called as "OCI" driver (Oracle Call Interface Driver)

=>These JDBC drivar s/ws are given in the form of jar files along with oracle Db s/w installation

classes111.jar -->oracle 8i
classes12.jar --> oracle 9i
ojdbc14.jar --> oracle10g
ojdbc6.jar -->oracle 11g
ojdbc7.jar --> oracle12c
.
.
.

Oracle thin driver details (Oracle corp supplied Type4 mechanism based JDBC driver
s/w for oracle)

jdbc drivor class name :: oracle.jdbc.driver.OracleDriver
pkg name JDBC drivor class name
implemennting java.sql.Driver(i)

jdbc url :: jdbc:oracle:thin:@<host>/<ipaddress>:<portno>:<sid>
eg1:: jdbc:oracle:thin:@localhost:1521:xe (for local oracle Db s/w)
refares to current computer..

eg2:: jdbc:oracle:thin:@34.67.89.666:1521:xe (for remote oracle Db s/w)
Ip address of
remote machine
where oracle db s/w
installaed

jdbc --main protocol
oracle --sub protocol
thin - sub name
localhost - current computer
where oracle is installed
1521 :: oracle s/w port number
xe :: sid (service id) (Logical Db name)

jar flex ::

classes111.jar -->oracle 8i
classes12.jar --> oracle 9i
ojdbc14.jar --> oracle10g
ojdbc6.jar -->oracle 11g
ojdbc7.jar --> oracle12c
.
.
.
C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar

Write JDBC App to establish the connection with Oracle Db s/w?

step1) keep the following s/w setup ready
=>jdk any version (jdk 1.8+)
=>oracle any version (oracle 10g +)
=>editplus /notepad ++

step2) add ojdbc6.jar (I have oracle11g) to CLASSPATH env.. variable..

mycomputer/This PC -->properties --> advanced system settings -->
env.. variable s ---> system variables --->
variable name :: CLASSPATH
value : C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6.jar;.
-->ok -->ok -->ok

ste3) Develop Java App to esablish the connection with oracle Db s/w using "oracle thin driver"
(Type4 JDBC driver s/w for oracle
given by oracle corp)

```
//ConnTest.java
import java.sql.*;

public class ConnTest
{
    public static void main(String[] args) throws Exception
    {
        //1.register JDBC drvier s/w
          //a. create JDBC driver class object
            oracle.jdbc.driver.OracleDriver obj=new oracle.jdbc.driver.OracleDriver();
            //register JDBC driver class object with DriverManager service
            DriverManager.registerDriver(obj);
    //esablish the connection
        Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                                                   "system","manager");
            if(con==null)
                System.out.println("connection is not established");
            else
          System.out.println("connection is established");

        }
}
//cmd>javac ConnTest.java
//cmd>java ConnTest
```

To collect SID
==========
SQL> select * from global_name;

GLOBAL_NAME

XE

```
                                                  JDBC URL
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                  static method                                  "system","manager");
                                                                  DB username    DB Pwd
```

DriverManager.getConnection(-,-,-) performs

1) Based on "jdbc:oracle:thin" info of the JDBC URL the getConnection(-,-,-) pickaup the registerted Oracle thin JDBC driver s/w from DriverManager service..

2) Locates "xe" sid based Logical DB in oracle Db s/w of current computer (localhost) which is running on port number 1521 and esblish the connection with that Oracle Db s/w.. In that process uses usess the given "system","manager" as DB username and password for authentication (security check)

3) getConnection(-,-,-)method returns JDBC Connection object (con) back to Java App representing connectivity between Java App and oracle Db s/w.

note: JDBC Connnection obj means ,it is the object of java class that implements java.sql.Connection().

We can write the following single line statement to register JDBC driver with DriverManager service insttread of the regular two lines of code

old code
========
```
oracle.jdbc.driver.OracleDriver obj=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(obj);
```

new code
==========
```
                                optional
Class.forName("oracle.jdbc.driver.OracleDriver");
                  (or)
Class.forName("oracle.jdbc.OracleDriver");
```

Q) In the Class.forName("oracle.jdbc.driver.OracleDriver") statement.. the JDBC driver class is just loaded and no object is created for jdbc driver class and registered with DriverManager service.. Then how can u say JDBC driver s/w registered?

Ans) Class.forName("oracle.jdbc.driver.OracleDriver") statement makes jvm to load given JDBC driver class "oracle.jdbc.driver.OracleDriver" .. In that process the static block of JDBC driver class executes.. In that static block logic is placed to create JDBC driver class obj and to register that JDBc Driver class obj with Driver Manager service by calling DriverManager.registerDriver(-) method .. Like this JDBC driver s/w is registered with DriverManager service..

static block of oracle.jdbc.driver.OracleDriver class looks like this
==========

```
static {
    try {
      if (defaultDriver == null) {
        defaultDriver = new oracle.jdbc.OracleDriver();   | register JDBc driver s/w
        DriverManager.registerDriver(defaultDriver);      | by creating JDBC driver class obj
      }
    } catch (SQLException se) {
        ...
    }
}//static
```

.java ---------javac----------------> .class | java compiler comes with jdk installation
(java compilation)

.class ------ decompiler----------------------> .java | we must arrange java decompilers seperately or we need to use on decompilers like decompiler.com
(java decompilation)

To see the source code of oracle.jdbc.driver.OracleDriver from ojdbc6/7/8.jar file.
==========

go to decompiler.com webisite ---> choose file ------>select ojdbc6/7/8.jar file ----> open oracle.jdbc, driver folders and see the source code of OracleDriver.java file..

JDBC4.x version feature that is auto-loading of JDBC driver class

If u r JDBC driver s/w is given based JDBC4 version then loading of jdbc driver class is not required.. If just added the JDBC driver s/w related jar file to the CLASSATH ..the JDBC driver classes will be loaded automatically.. based on the JDBC URL passed in the DriverManager.getConnection(-,-,-) method.

=>ojdbc14.jar represents oracle thin driver (JDBC3 version)
(No support for autoloading JDBC driver class)
(HEre we need to load JDBC driver class explicitly using Class.forName(-) method)

=>ojdbc6/7/8.jar represents oracle thin driver (JDBC 4 Version)
(Supports autoLoading JDBC driver class)
(HEre we need not to load JDBC driver class explicitly using Class.forName(-) method)

Example App to establish the connection with Oracle Db s/w using JDBC4 based Oracle thin Driver

step1) make sure that ojdbc6/7/8.jar file is added to CLASSPATH
note:: from oracle11g all the oracle thin drivers that are coming along with oracle installation are based JDBC4 Version supporting auto Loading JDBC driver class.

step2) Develop Application as shown below

ConnTest.java

```
public class ConnTest
{
    public static void main(String[] args) throws Exception
    {
    //2.esablish the connection
        Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                                                   "system","manager");
        If(con==null)
            System.out.println("connection is not established");
        else
    System.out.println("connection is established");
    }
}
//cmd>javac ConnTest.java
//cmd>java ConnTest
```

How does this autoLoading JDBC driver class is happening?

In ojdbc6/7/8.jar file
|----> META-INF
|---->services
|-----> java.sql.Driver (file)
oracle.jdbc.OracleDriver (jdbc driver class)

when DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system","manager"); is called the static

block of DriverManager class executes and this static block logic will be there to search for java.sql.Driver file in META-INF/services folder of all the jar files that are added to CLASSPATH.. and finds in ojdbc6/7/8.jar file.. and gets oracle.jdbc.ORacleDriver class from that file (java.sql.Driver file)

and loads that jdbc driver class automatically .. (In this process the static block of JDBC driver class exececutes and JDBc driver s/w will be regstried with DriverManager service automatically)

note: In ojdbc14.jar file there is no java.sql.Driver file in any folder.. So the ojdbc14.jar file (JDBC3) does not support auto loading of JDBC driver class..

Date d=new Date();
S.o.println(d.hashCode());

=>Gives the hashCode of the object
=>hashCode() is the public of java.lang.Object class..
=>since java.lang.Object is top most class in the inheritence hierarchy of every java class ... we can call public methods that class on any java class object.

S.o.println("class name of d obj is "+d.getClass());

=> It is also public method java.lang.Object class .. which gives class name of given object..

//2.esablish the connection (class name)
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
interface static method "system","manager");
System.out.println("con object class name :"+con.getClass());

Gives JDBC con object class name .. whose name changes based on JDBC driver s/w we are using...
in Oracle thin driver s/w . the JDBC con object classname is " oracle.jdbc.driver.T4CConnection"
which impelements java.sql.Connection() indirectly..

Q) If java.sql.Connection is an interface , then how can u say DriverManager.getConnection(-,-,-) method is returning JDBC Connection object?

Ans1) JDBC con object means it is not the object of java.sql.Connection() (we can not create obj for interfaces) . It is the object of underlying JDBC driver s/w supplied Java class that implements java.sql.Connection(). Since the JDbc con object class name changes JDBC driver s/w to JDBC driver s/w .. we never refer that jdbc con object with real class name reference variable. we always refer with the commonly implemented java.sql.Connection() reference supporting interface based Runtime polymorphism and loose coupling

//2.esablish the connection
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system","manager");
System.out.println("con object class name :"+con.getClass());

Gives JDBC con object class name .. whose name changes based on JDBC driver s/w we are using...
In Oracle thin driver s/w . the JDBC con object classname is " oracle.jdbc.driver.T4CConnection"
which impelements java.sql.Connection() indirectly..

Ans2) =>List object does not mean object of java.util.List(). It is the object of a class that implementss java.util.List() (like ArrayList object, Vector object ,Stack Object and etc..)

=> Serializable object doest not mean object of java.io.Serializable().. It is the object of a java class that implements java.io.Serializable().

=> Cloneable object doest not mean object of java.lang.Cloneable().. It is the object of a java class that implements java.lang.Cloneable().

=>Similary JDBC Con object means .. It is not the object of java.sql.Connection().. It is the object of underlying JDBC driver s/w supplied java class that implements java.sql.Connection().

note:: In any programming , If the object name is refered using interface name .. that does not mean object is created for interface .. It is refering object created for implementation of class of that interface.

Ans3) Signature of DriverManager.getConnection(-,-,-)method

public static Connection getConnection(String jdbcUrl,
(interface) String dbuser,
String dbpwd)throws SQLException

=> since the return type java.sql.Connection() .. The DriverManager.getConnection(-,-,-) method returns java.sql.Connection() impl class object as jdbc con object back to Java App representing connectivity b/w Java App and DB s/w.

3 imp statements on method return type

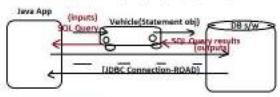
a) If java method return type is an interface, then method returns one of its impelementation class obj as the return value.
b) if java method return type is an abstract class, then method returns one of its sub class as the return value.
b) if java method return type is concrete class, then method can return either that concrete class obj or one of its sub class object.

note: IN JDBC programming we get
statement obj
ResultSet obj
PreparedStatement obj
and etc..

=>All these are not objects of interfaces beloging to JDBC API.. all these are objects of implementation classes implementing jdbc api interfaces .

JDBC Statement object

=>JDBC STatement object acts vechicle b/w Java App and Db s/w to carry inputs (SQL queries) toDb s/w from Java App and to carry output (SQL query results) back to Java App from DB s/w.



To create Statement obj

Statement st=con.createStatement();

System.out.println("Statement object (st) class name::"+st.getClass());

in oracle thin driver statement obj class name is oracle.jdbc.driver.OracleStatementWrapper (c) Implementing java.sql.Statement(I) directly or indirectly.

when java.sql.Statement is an Interface ,how can say con.createStatement() method returns JDBC Statement object?

Ans) It is not the object of java.sql.Statement(I). It is the object of underlying JDBc driver s/w supplied java class that implements java.sql.Statement(I). Since class names will change based on the JDBC Driver s/w and DB s/w we use... we never specify real class name to refer JDBC statement object .. More over we always refer with common interface refer variable that is java.sql.Statement (I) ref variable (st).

According Oracle Specification there are 4 types of SQL queries

a) DML Queries (Data Manipulation Language Queries)
   -> insert, update and delete SQL queries
b) DRL Queries (Data Retrieving Language Queries)
   -> select SQL queries
c) DDL Queries (Data Definition Language Queries)
   ->create table, drop table, alter table and etc..
d) TCL Queries (Transaction Control languages queries)
   -> commit , rollback, savepoint and etc..

Accoding JDBC /JAva there are two types SQL queries to send and execute in DB s/w

a) SELECT SQL queries

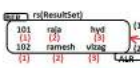
=>DRL Queries (These queries give bunch of records that are selected Db s/w after executing the SQL query)

=>use executeQuery(-) on Statement object to send and execute SQL Select Queries to DB s/w

=> executeQuery(-) returns JDBC ResultSet obj representing the records given by SELECT SQL execution.

Statement st=con.createStatement();

eg:: ResultSet rs= st.executeQuery("SELECT * FROM STUDENT");



BFR->Before First Record
ALR ->After Last Record

Student (db table)

sno(n)	sname(vc2)	sadd(vc2)
101	raja	hyd
102	ramesh	vizag

=>The records in ResultSet obj and the col values in each record of ResultSet obj will be maintained having (1) based index.
(One based index)

=>SELECT SQL query execution gives bunch of records in the form of ResultSet, So executeQuery(-) method return type is ResultSet

public ResultSet executeQuery(String query)throws SQLException

b) NON-SELECT SQL queries

=>DML Queries, DDL Queries , TCL Queries

=> DML queries return numberic value representing no.of records that are effected after query eexecution.

=>To send and execute non-select SQL Query (especially DML Query) we can use excouteUpdate(-) on JDBC statment obj.

=> executeUpdate(-) method returns numeric value back to Java App representing no.of records that are effected becoz of non-select SQL query (DML) execution.

int count1=st.executeUpdate("DELETE FROM STUDENT WHERE SNO>=100");

gives 2, if two records are deleted..
gives 0, if no records are deleted.

=> Non-select SQL Query(DML Query) return numberic value representing no.of records that are effected so the executeUpdate(-) return type is "int".

public int executeUpdate(String query)throws SQLException.

note:: SQL keywords , DB table names, col names are not case-sensitive in SQL .. but col data/values are case-sensitive

Q) How can we get default Db tables in oracle ?

=>Go to oracle installation folder --> search form scott.sql and copy and paste line 28 to till end from scott.sql to SQL prompt (SQL plus)

Write a JDBC App to get records from "STUDENT" Db table of oracle Db s/w and display them on the console

step1) keep the following s/w setup ready

=>oracle any version (11g)
=> java version (jdk 1.8+)
=> EditPlus any version

(same as First App setup)

step2) Launch SQL plus or sql prompt window and connect to Oracle Db s/w.

=>Launch "Run commandline" from Oracle installation
SQL> connect system/manager
Connected.

step3) create db table "STUDENT" having records..

Primary key column (PK) = NonNull + Unique column

SQL> create table student (sno number(5) primary key,sname varchar2(20),sadd varchar2(20),avg number(10,2));
Table created.

SQL> desc student;

Name	Null?	Type
SNO	NOT NULL	NUMBER(5)
SNAME		VARCHAR2(20)
SADD		VARCHAR2(20)
AVG		NUMBER(10,2)

SQL> select * from tab; (To get all the db tables)

SQL> Insert into student values(101,'raja','hyd',90.66);
1 row created.
SQL> Insert into student values(102,'rajesh','delhi',91.66);
1 row created.
SQL> Insert into student values(103,'suresh','mumbai',92.66);
1 row created.
SQL> commit;
Commit complete.

SQL>select * from student;

SQL> select * from student;

SNO	SNAME	SADD	AVG
101	raja	hyd	90.66
102	rajesh	delhi	91.66
103	suresh	mumbai	92.66

step3) Develo the Java App having JDBC code to get the records "STUDENT" db table.

JDBC Statement object
=====================

=>JDBC STAtement object acts vechicle b/w Java App and DB s/w to carry inputs (SQL queries) to DB s/w from Java App and to carry output (SQL query results) back to Java App from DB s/w.

To create Statement obj
=======================

Statement stmt=con.createStatement();

System.out.println("Statement object [str] class name:"
+stmt.getClass());

In oracle 10xe driver statement obj class name is
oracle.jdbc.driver.OracleStatementWrapper (v.)
Implementing java.sql.Statement() directly or indirectly.

when java.sql.Statement is an interface ,how can say con.createStatement() method returns JDBC Statement object?

Ans) It is not the object of java.sql.statement(). It is the object of underlying JDBc driver s/w supplied java class that implements java.sql.Statement(). Since class names will change based on the JDBC Driver s/w and DB s/w we use.. we never specify real class name to refer JDBC statement object .. More over we always refer with common interface refer variable that is java.sql.Statement () ref variable (st).

According Oracle Specification there are 4 types of SQL queries

a) DML Queries [Data Manipulation Language Queries]
=> insert, update and delete SQL queries
b) DRL Queries [Data Retrieving Language Queries]
=> select SQL queries
c) DDL Queries [Data Definition Language Queries]
=> create table, drop table, alter table and etc...
d) TCL Queries [Transaction Control languages queries]
=> commit , rollback, savepoint and etc...

According JDBC /1Aua there are two types SQL queries to send and execute in DB s/w

a) SELECT SQL queries
=>DRL Queries ((Thase queries give bunch of records that are selected DB s/w after executing the SQL query)
=>use executeQuery() on Statement object to send and execute SQL Select Queries to DB s/w
=> executeQuery() returns JDBC ResultSet obj representing the records given by SELECT SQL execution.

```
Statement stmnos.createStatement();
eg:- ResultSet rs =s.executeQuery("SELECT * FROM STUDENT");
```

Student (db table)

sno(n)	sname(vc2)	sadd(vc2)
101	raja	hyd
102	ramesh	vizag

rs(ResultSet)

101	raja	hyd	(1)
(1)	(2)	(3)	
102	ramesh	vizag	(4)
(4)	(5)	(6)	new

BRS->Before First Record
ALR ->After Last Record

=>The records in ResultSet obj and the col values in each record of ResultSet obj will be maintained having (1) based index.
(Zero based index)

=>SELECT SQL query execution gives bunch of records in the form of ResultSet, So executeQuery() method return type is ResultSet

```
public ResultSet executeQuery(String query)throws SQLException
```

b) NON SELECT SQL queries

=>DML Queries, DDL Queries , TCL Queries

=> DML queries return numberic value representing no.of records that are effected after query execution.

=>To send and execute non-select SQL Query (especially DML Query) we can use executeUpdate() on JDBC statement obj.

=> executeUpdate() method returns numeric value back to Java App representing no.of records that are effected becoz of non-select SQL query (DML) execution.

```
int xcount1=st.executeUpdate("DELETE FROM STUDENT WHERE SNO>=100");
```

gives 2, if two records are deleted..
gives 0, if no records are deleted.

=> Non-select SQL Query(DML Query) return numberic value representing no.of records that are effected so the executeUpdate() return type is "int".

```
public int executeUpdate(String query)throws SQLException;
```

note:- SQL keywords , DB table names, col names are not case-sensitive in SQL .. But col data/values are case sensitive

Q) How can we get default db tables in oracle ?
=>Go to oracle installation folder --> search form scott.sql and copy and paste like 26 to till end from scott.sql to SQL prompt (SQL plus)

Write a JDBC App to get records from "STUDENT" Db table of oracle DB s/w and display them on the console

step1) keep the following s/w setup ready

=>oracle any version (11g)
=> java version (jdk 1.8+)
=> IDEPNot any version

[same as First App setup]

step2) Launch SQL plus or sql prompt window and connect to Oracle Db s/w.

=>Launch "Run command Box" from Oracle installation
SQL> connect system/manager
Connected.

step3) create db table "STUDENT" having records..

Primary key column (PK) = NonNull + Unique columns

SQL> create table student (sno number(3) primary key,sname varchar2(20),sadd varchar2(20),avg number(10,2));
Table created.

```
SQL> desc student;
Name                      Null?    Type
------------------------- -------- ----------
SNO                       NOT NULL NUMBER(3)
SNAME                              VARCHAR2(20)
SADD                               VARCHAR2(20)
AVG                                NUMBER(10,2)
```

SQL> select * from tab; [To get all the db tables]

```
SQL> insert into student values(101,'raja','hyd',90.66);
1 row created.
SQL> insert into student values(102,'rajesh','delhi',91.66);
1 row created.
SQL> insert into student values(103,'ramesh','mumbai',92.66);
1 row created.
SQL> commit;
 Commit complete.

SQL>select * from student;
```

```
SQL> select * from student;

       SNO SNAME           SADD            AVG
---------- --------------- --------------- ----------
       101 raja            hyd                  99.66
       102 rajesh          delhi                91.66
       103 ramesh          mumbai               92.66
```

Step4) make sure that ojdbc14/6/7.jar file is added to CLASSPATH env. variable.
(control panel --> system properties --> Advanced --> Environment variables) (same as first App)

step5) Develop the Java App having JDBC code to get the records "STUDENT" db table.


```
while( rs.next() ==true)
    System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(4));
 }
```

```
console
=======
101  raja  hyd  90.66
102  rajesh  delhi  91.66
103  ramesh  mumbai  92.66
```

=>rs.next() method moves the runner in ResultSet object from current position to next to position.. If that next position is record then this method returns "true" otherwise (if that next position is ALR or BFR not a record) then it returns false.

=> Even though ResultSet object is created as empty object (no records)...Once also the ResultSet object contains BFR , ALR positions.
=> To get col values from the records of ResultSet...we can use getXxx() methods on ResultSet obj either passing col indexes or col names.

The getXxx() methods are
getInt() , getString() ,getFloat() ,getDouble() ,getLong() and etc... (15+ methods are there...)

=>All JDBC objs are internally IO Stream, network socket objs ...So it is recommended to close them at the end of their utilizations that is in reverse order of opening]

```
rs.close()
st.close()
con.close();
```

=>closing jdbc connection (con.close()) indirectly closes statement, ResultSet objs also.
=> It is recommended to close jdbc con object at the end of utilization to avoid the problem DB s/w reaching max connections and not giving connections to new client Apps (becoz already opened connections with DB s/w are not closed and they are in idle mode.

note:-while processing the ResultSet .. If u do not know col data types...the we can get all the col values as String values using rs.getString() methods either passing col names or col indexes.

```
//process the ResultSet object
while(rs.next()==false)  // while(rs.next()==true)
  System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(4));
                    (or)
  System.out.println(rs.getInt("SNO")+"  "+rs.getString("SNAME")+"  "+rs.getString("SADD")+"  "+rs.getFloat("AVG"));
                    (or)
  System.out.println(rs.getString("SNO")+"  "+rs.getString("SNAME")+"  "+rs.getString("SADD")+"  "+rs.getString("AVG"));
                    (or)
  System.out.println(rs.getString(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getString(4));
}
                    NOT
note:-By default JDBC Apps can read sequential of DB s/w..
```

Code
====

//SelectTest.java

```
import java.sql.*;

public class SelectTest
{
  public static void main(String args[])throws Exception{

        //Load jdbc driver class
          // Class.forName("oracle.jdbc.driver.OracleDriver");

    //establish the connection (Read)
          Connection con=DriverManager.getConnection("jdbc:oracle:thin@localhost:1521:xe",
                                                     "system","manager");
        //create JDBC Statement object (vechicle)
          Statement stmnos.createStatement();
        //Send and execute SQL SELECT Query in DB s/w and get JDBC ResultSet object
          ResultSet rs= st.executeQuery("SELECT * FROM STUDENT");

            //process the ResultSet object
            while(rs.next()==false)  // while(rs.next()==true)
                    //System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(4));
                    //System.out.println(rs.getInt("SNO")+"  "+rs.getString("SNAME")+"  "+rs.getString("SADD")+"  "+rs.getFloat("AVG"));
                    //System.out.println(rs.getString("SNO")+"  "+rs.getString("SNAME")+"  "+rs.getString("SADD")+"  "+rs.getString("AVG"));
                    System.out.println(rs.getString(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getString(4));


                    }

                    //close jdbc objs
                     rs.close();  //optional
                     st.close();  //optional
                     con.close();  //mandatory
    }
}
```

```
cmd> javac  SelectTest.java
cmd> java  SelectTest
```

Assignment:
===========

Write a JDBC App to read and display records from "EMP" default db table of Oracle DB s/w

After installing Oracle Db s/w, make sure that Oracle related two services in running mode

serach for services.msc ----> In that search for the following two services and make sure that they in running mode (if not u tart them manually)

Soft Coding vs Hard Coding

int a=10; //hard coding
(Typing the input value directly in java source code)

=>Softcoding is recemended

Scanner scnew Scanner(System.in)
int a=sc.nextInt(); //soft coding
(Collecting values from enduser with out typing them in directly in the source code of the App..

=>The standard slogan in s/w industry is
Do not hard code inputs in a r Java app
that are changable in future ...always get from outside the App through softcoding .

Diffest ways of softcoding in Java

(standalone Apps)
a) cmd line args (using args[0], args[1]....) | (from java 1)
b) System properties (using System.getProperty(-)) | (from java 1)
c) Scanner (from java5) (best)
d) Console (from java 6)
e) Using io Streams (Like BufferReader) | (java1)
f) Awt/swing GUI comps (text boxes , select box , radio buttons and etc...) | AWT--> java1 , swings -->java5)
g) Using Html form comps (in web applications)
(In support with servlet, jsp technologies)

Write a JDBC App to empno,ename, job,sal col values from based on the given salary range (get min and max salaryies from Enduser using softcoding [Scanner])

```
SQL> desc emp;
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                     NOT NULL NUMBER(4)
 ENAME                              VARCHAR2(10)
 JOB                                VARCHAR2(9)
 MGR                                NUMBER(4)
 HIREDATE                           DATE
 SAL   (6)                          NUMBER(7,2)
 COMM                               NUMBER(7,2)
 DEPTNO                             NUMBER(2)

SQL> SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>=3000 AND SAL<=5000;
                  retciviing specific col values          (get these values throuhg soft coding)
     EMPNO ENAME      JOB              SAL
---------- ---------- ------------- ---------
      7788 SCOTT      ANALYST          3000
      7839 KING       PRESIDENT        5000
      7902 FORD       ANALYST          3000
```

=> Technically ";" symbol is not part of SQL Query .. So while sending SQL query to DB s/w from any Front End App (like Java App , .net App , PHP App and etc..) There is no need of keeping ";" symbol at end of the SQL query .. If is just required in SQL prompt or SQL Plus to tell to SQLPrompt that the SQL query is ended.

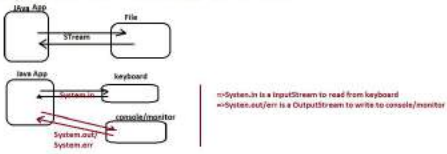
```
SQL> SELECT * FROM STUDENT;

In JDBC App
===========
ResultSet rsrst.executeQuery(" SELECT * FROM STUDENT ");
```

3 Built in streams of Java App

=> System.in (Standard Input stream pointing to keyboard)
=> System.out (Standard output stream pointing to console/monitor) (Use it for out put messages)
=> System.err (standard error/output stream pointing to console/monitor) (Use it for error messages)

[in ,out, err are static properties of System class]

=>Stream is continuous flow of data.. that representing connectivity b/w Soruce and destination.



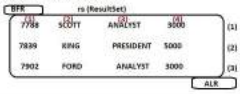
=>System.in is a InputStream to read from keyboard
=>System.out/err is a OutputStream to write to console/monitor

Can u give sample JDBC URL that have used in u r Project?

Fresher :: jdbc:oracle:thin:@localhost:1521:xe
or
jdbc:oracle:thin:@localhost:1521:orcl

Exprerienced :: jdbc:oracle:thin:<host>:<1521>:<sid>
sample :: jdbc:oracle:thin:pvpd109.db.aexp.com:1521:pvss109p"
company specific host name example
company specific sid example
U can collect this info in the company either DBA or TL/PL

```
//prepare SQL query
  //SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>=1000 AND SAL<=3000
  String query="SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>="+startSalary+" AND SAL<="+endSalary;
                                          1     2     3    4
  System.out.println(query);
 //send and execute SQL Query in Db s/w
 ResultSet rs=st.executeQuery(query);
```



=>While processing the ResultSet , we should give col indexes in the order they selected and stored in ResultSet obj .. Not in the order columns are there in DB table.

Example Code (SelectTest1.java)

```
//SelecctTest1.java

import java.sql.*; // for jdbc api
import java.util.*; // for Scanner

public class SelectTest1
{
    public static void main(String[] args) throws Exception
    {
        //read inputs
          Scanner scrnew Scanner(System.in);
          System.out.println("Enter start range of employee Salary::");
          float startSalary=sc.nextFloat(); //gives 1000
          System.out.println("Enter start range of employee Salary::");
          float endSalary=sc.nextFloat(); //gives 3000

          //register JDBC driver s/w by load jdbc Driver class (optional)
            //Class.forName("oracle.jdbc.driver.OracleDriver");

    //establish the Connection with Oracle Db s/w
              Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
              //create Statement object
              Statement st=con.createStatement();
                //prepare SQL query
          //"SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>=1000 AND SAL<=3000
        String query="SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>="+startSalary+" AND SAL<="+endSalary;
          System.out.println(query);
              //send and execute SQL Query in Db s/w
                ResultSet rsrst.executeQuery(query);
                //process the ResultSet obj
          System.out.println("Emp details having salary range "+startSalary+" ..."+endSalary);
                        while(rs.next()!=false){
                                System.out.println(rs.getInt(1)+"  "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                            }
        //close jdbc objs
          rs.close();
          st.close();
          con.close()
    }//main
}//class
//=javac SelectTest1.java
//=java SelectTest1
```

```
G:\Workspaces\advjava\NTAH15\JDBC>javo SelectTest1
Enter start range of employee Salary::
3000
Enter start range of employee Salary::
5000
SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>=3000.0 AND SAL<=5000.0
Emp details having salary range 3000.0 ...5000.0
7788 SCOTT ANALYST 3000.0
7839 KING PRESIDENT 5000.0
7902 FORD ANALYST 3000.0

G:\Workspaces\advjava\NTAH15\JDBC>java SelectTest1
Enter start range of employee Salary::
6000
Enter start range of employee Salary::
10000
SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE SAL>=6000.0 AND SAL<=10000.0
Emp details having salary range 6000.0 ...10000.0
```

Assignment:: Write a JDBC App to get Student details from STUDENT DB table based on the given avgerages range ?

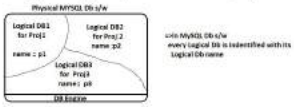
Write a JDBC App to get Dept details from DEPT DB table based given dept number range?

MYSQL:

=>type :: RDBMS DB s/w
=>version :: 8.x
=>Vendor :: Devx/ SunMs/ Oracle corp
=>Free DB s/w (open source)
=>default port no :: 3306
=> default username :: root
=> default password :: not fixed .. should be chosen installation (i prefer root)
=> gives multiple default Logical DBs like test,mysql,ukalla and etc..

URL to download MYSQL DB s/w
https://dev.mysql.com/downloads/mysql/

Logical Db is a logical partision of physical Db s/w.. and it will be created on 1 per Project basis.



=>in MySQL Db s/w
every Logical Db is indentified with its
Logical Db name

Procedure to create Logical DB having DB table with records in mysql DB s/w

step1) Make sure that MYSQL DB s/w is installed in the Computer

step2) launch MYSQL SQL prorrpt or Command Line prorrpt

search for mysql ---> and launch MYSQL Command line client
passoword : root

step3) create Logical DB

```
mysql> create database NTAJ415DB;
Query OK, 1 row affected (0.24 sec)

mysql> connect NTAJ415DB;
Connection id:  9
Current database: NTAJ415DB
```

step4) create DB table STUDENT and insert records into it

```
mysql> create table student(sno int(5) primary key,sname varchar(20),sadd varchar(20),avg float);
Query OK, 0 rows affected, 1 warning (1.98 sec)

mysql> insert into student values(101,'raja','hyd',89.55);
Query OK, 1 row affected (0.12 sec)

mysql> insert into student values(102,'rajesh','vizag',59.55);
Query OK, 1 row affected (0.13 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)


mysql> select * from student;
+-----+--------+-------+-------+
| sno | sname  | sadd  | avg   |
+-----+--------+-------+-------+
| 101 | raja   | hyd   | 89.55 |
| 102 | rajesh | vizag | 59.55 |
+-----+--------+-------+-------+
2 rows in set (0.08 sec)
```

note:: Devx has released same mysql DB s/w with the name MariaDB.

To get all existing logical DBs of mysql Db s/w

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| ntaj114db1         |
| ntaj414db          |
| ntaj414db1         |
| ntaj415db          |
| nth0415db          |
| ntqk134b           |
| ntsp714db          |
| ntspbrra714db      |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
14 rows in set (0.46 sec)
```

default logica DBs

default logica DB

default logica DBs

=>Devx supplied type4 mechanism based JDBC driver s/w for mysql is called Connector/J JDBC driver and its jar file will not come along with MYSQL DB s/w installation.. we should download it seperately.

mysql-connector-java-<ver>.jar --> connector/J JDBC driver s/w jar file
(url :: https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.25)
(Supports autloading of JDBC driver class)
becoz it is having
META-INF/services
|--> java.sql.Driver (file)
com.mysql.cj.jdbc.Driver
jdbc driver class name Implementing java.sql.Driver(I)

Connector/J JDBC driver details

mechanism :: type4
target DB s/w :: Mysql
JDBC driver class name :: com.mysql.cj.jdbc.Driver
JDBC url :: jdbc:mysql:///<LogicalDB> (For Local mysql DB s/w i.e java App and mysql DB s/w are there on same machine)
jdbc:mysql://<host>or<ipaddrss>:<portno>/<Logical DB> (For remote mysql DB s/w i.e Java App and mysql DB s/w are there in two different machies)

eg: jdbc:mysql://localhost:3306/NTAJ415DB

protocol | host name and port no of mysql DB s/w | Logical DB

The DriverManager.getConnection(...) picks the one JDBC Driver s/w from multiple registerted JDBC driver s/ws based on protocol info of JDBC URL

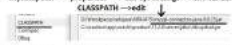
jdbc:oracle:thin of JDBC URL--> picks up oracle thin driver
jdbc:mysql of JDBC URL--> picks up mysql connect/J driver
jdbc:postgres of JDBC URL--> picks up postgresql JDBC driver

jar file :: mysql-connector-java-<ver>.jar
(mysql-connector-java-8.0.25.jar)
(download from internet seperately)

Procedure to Java App interacting with MYSQL DB s/w using connector/J JDBC driver s/w

step1) make sure that mysql-connector-java-8.0.25.jar file is added CLASSPATH env.. variables

mycomputer --->properties ---> adv sys settings -->env variables --> system variablee-->
CLASSPATH --->edit



step2) make sure that DB table with records is availble in mysql Db s/w.. (refer above)

step3) gather mysql Connector/J JDBC driver details (refer above)

step4) Develop the JDBC App

//SelectTest_MySQL.java

```
import java.sql.*;

public class SelectTest_MySQL
{
  public static void main(String args[])throws Exception{

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

      //establish the connection (Road)
        //Connection con=DriverManager.getConnection("jdbc:mysql:///NTAJ415DB","root", "root");
          Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/NTAJ415DB","root", "root");

          //create JDBC Statement object (vechicle)
            Statement st=con.createStatement();
          //Send and execute SQL SELECT Query in Db s/w and get JDBC ResultSet object
            ResultSet rs= st.executeQuery("SELECT * FROM STUDENT");

            //process the ResultSet object
              while(rs.next()!=false){  // while(rs.next()==true)
                  System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+ rs.getString(3)+"  "+rs.getFloat(4));
              }

              //close jdbc objs
               rs.close();  //optional
               st.close();  //optional
               con.close();  //mandatory
    }//main
}//class
//cmd>javac SelectTest_MySQL.java
//cmd>java SelectTest_MySQL
```

```
G:\Workspaces\advjava\NTAJ415\JDBC>javac SelectTest_MySQL.java

G:\Workspaces\advjava\NTAJ415\JDBC>java SelectTest_MySQL
101  raja  hyd  89.55
102  rajesh  vizag  59.55
```

note:: The class names of JDBC objects will change based on JDBC driver s/ws and DB s/ws we use ...
So we never refer then using real class names ref varaibles.. becoz that makes our code tightly coupled code.. We always refer them using common jdbc api interfaces ref variable names which gives loose coupling...

while working with Oracle thin Driver

```
System.out.println("con object class name::"+con.getClass());
System.out.println("st object class name::"+st.getClass());
System.out.println("rs object class name::"+rs.getClass());
```

```
con object class name::class oracle.jdbc.driver.T4CConnection
st object class name::class oracle.jdbc.driver.OracleStatementWrapper
rs object class name::class oracle.jdbc.driver.OracleResultSetImpl
```

While working with Connector/J Driver of mysql

```
System.out.println("con object class name::"+con.getClass());
System.out.println("st object class name::"+st.getClass());
System.out.println("rs object class name::"+rs.getClass());
```

```
con object class name::class com.mysql.cj.jdbc.ConnectionImpl
st object class name::class com.mysql.cj.jdbc.StatementImpl
rs object class name::class com.mysql.cj.jdbc.result.ResultSetImpl
```

Java Coding standards

(1) Following Java Naming Conventions
class /Interface/enum/Annotation :: every word first letter must be capital
eg:: SelectTest , EmployeeManagementServiceLocator and etc..
=> method names/varibles name :: First word-> all are lower letters
and second onwards first letter is upper case letter
=> registerDriver(-) , getConnection(-,-,-) , processData(-), listEmployees(-) and etc...
=> int personAge , float studentAvg , int sno and etc..

(2) Take names(class/interface/enum/annotation/method/variable) in programming having self deescription.
int a=10; //storing age ✗
int personAge=10 //strong age ✓

(3) Avoid implicit package imports and prefer explicit package imports

```
import java.sql.*;  //implicit pkg imports  (not good)
import java.util.*;
```
while using classess of the pkgs in App we can not say which class belongs which pkg (not readility in code)

```
import  java.sql.DriverManager;  //explicit pkg import  (good)
import  java.sql.SQLException;
import  java.sql.Connection;

import  java.util.Scanner ;
```
[We can indentify , which class belongs to which package easily i.e improves the readability the the code]

(4) Write proper comments while writing the code.

(5) Do not delcare the exception to be thrown using "throws" in standalone Apps (class with main(-)) or single layer Apps .. always catch and handle exceptions usin try/catch blocks or try with resource (from java7)

(6) Followed by try block do not one catch (-) block handling all exception with Exception/Throwable as arguments place multiple catch handling multiple exception.. Initial catch blocks to handle known exceptions.. and use last catch block with Excption/Throwable argument to handle known Exceptions.

```
try{
  ....
  ...  //jdbc code.
  ...
}
catch(SQLException se){
  ...
}
catch(Exception e){
  ...
}
```

In Projects code will be placed catch(-) to display exception related messages as non-technical guding messages based on the exception that is raised..

Using getErrocode() of SQException we can get error code related SQL Query execution and we display message accordingly..

(7) Do not close jdbc objs , stream objs at the end of try block or in the catch block .. prefer closing them in finally block becoz this block irrespective exception raised in try block ..

```
try{
  ...
  ...  //jdbc code
  ...
  ...
}
catch(SQLException se){
  ...
}
catch(Exception e){
  ...
}
finally{
  ...
  ...  //close jdbc  objs
  ..
}
```

note:: try with resource can also be used..

(8) Follow proper indentation (tar spaces)

```
public class SelectTest{
    public  static void main(String args[]){
        Scanner  sc=new Scanner(System.in);
          ...
          ...
        }
}
```

Most of the times IDEs takes care indentation..

(9) Every user-defined class/Interface/enum/annotation must be delared in a package

```
public class SelectTest{
  ....
}
```

```
package com.nt.jdbc;
public class SelectTest{
  ...
}
```

=>it recomanded to create packge names in lower case letters having multiple words (minimun 3 words) with self description

notation1:: <companyname>.<projectname>.<category>;
eg:: nit.openfx.jdbc; ✓ (2)

notation2:: <com/org>.<projectname>.<module name>.<category>;
eg:: com.nit.library.jdbc; ✓ (1)

notation3:: <country name> . <company name>.<category>;
eg:: in.nit.jdbc;

notation3:: <country name> . <person name>.<category>;
eg:: in.natraj.jdbc;

(10) To avoid "NullPointerException" in the execution of the App , we must check for not null before calling/invoking methods or variables on that reference variable.

```
Statement st=null;
If(con!=null)
    st=con.createStatement();

ResultSet rs=null;
If(st!=null)
  rs=st.executeExecuteQuery("SELECT * FROM STUDENT");
```

if call/invoke method /variable on reference variable when it is having "null" then we get NullPointerException to avoid it go for not nul cehck.

In java variable declared inside block {....} is not visible outside block .. To overcome this problem declare the vairable before block and use inside and out side(after) the block.

=> In Java No default values for Local variables .. they must be initialized during the declaration.

->initialize ref variables with "null".
-> initialize int ,long,short,byte variables with "0".
> initialize float ,dpouble variables with "0.0f", "0.0" respectively
> initialize boolean variables with "false".

11) talk constant variables (private/public static final variable) in UPPECASE letter seperating multiple words with "_" symbol..

eg:: private/public static final float PI=3.14f;

```
private/public  static  final  int  MAX_VALUE=10;
private/public  static  final  int  MIN_VALUE=0;
```

12) Recomanded to write SQL Queries in the App in upper letters

and etc...

Download Java coding standards document from internet and get more....standards..

Write a JDBC App to get Employee Details from EMP DB table based on the given 3 desgs

```
SQL> select empno,ename,job,sal,deptno from emp where job in('CLERK','MANAGER','SALESMAN') order by job;
```

```
//SelectTest2.java
package com.nt.jdbc;
/* JDBC App that gets employee details from EMP DB table based on given 3 designations
   version :: 1.0
      author :: Team-J
*/
import java.util.Scanner;
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;


public class SelectTest2
{
      public static void main(String[] args)
      {
            //read inputs from enduser
            Scanner sc=null;
            String desg1=null,desg2=null,desg3=null;
             Connection con=null;
             Statement st=null;
             ResultSet rs=null;
            try{
        sc=new Scanner(System.in);
            if(sc!=null){
                  System.out.println("enter desg1:::");
                  desg1=sc.next().toUpperCase(); //gives CLERK
                  System.out.println("enter desg2:::");
                  desg2=sc.next().toUpperCase(); //gives MANAGER
                  System.out.println("enter desg3:::");
                  desg3=sc.next().toUpperCase(); //gives SALESMAN
                  }//if
       //covert input values as required for SQL query
        desg1="'"+desg1+"'";  //gives 'CLERK'
                  desg2="'"+desg2+"'";  //gives 'MANAGER'
                  desg3="'"+desg3+"'";  //gives 'SALESMAN'

                  //load JDBC driver class
                    //Class.forName("oracle.jdbc.driver.OracleDriver");

        //establish the connection
                    con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

                    //create Statement object
                    if(con!=null)
                     st=con.createStatement();

                    //prepare SQL query
                        //select empno,ename,job,sal,deptno from emp where job in('CLERK','MANAGER','SALESMAN') order by job
          String query="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE JOB IN("+desg1+","+desg2+","+desg3+") ORDER BY JOB";
                          // ctrl+u :: to get selected content in uppercase  ctrl+l :: to get selected content in lowercase
                            System.out.println(query);

       //send and execute SQL Query in DB s/w
                            if(st!=null)
                                rs=st.executeQuery(query);

                            if(rs!=null){
                                System.out.println("The Employee details are ");
              while(rs.next()!=false){
                System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(6)+"  "+rs.getInt(5));
                                }//while
                            }//if
            }//try
            catch(SQLException se){  // To handle known exception
          System.out.println(se.toString());  // gives details info about the raised exception
            }
            catch(Exception e){  //To handle unkwon exceptions
        e.printStackTrace();
            }
        /*  finally{
                    //close jdbc objs & stream objects
                    try{
                        if(rs!=null && st!=null && con!=null && sc!=null){
              rs.close();
                            st.close();   //Bad code becoz of while closing one object
                            con.close(); //If any exception raised the control goes catch(-) does not
                            sc.close();  // give fair chance for the objects in try block to get closed
                          }//if
                        }//try
                        catch(SQLException se){
                            se.printStackTrace();
                        }
                        catch(Exception e){
                            e.printStackTrace();
                        }
                    }//finally */

                    finally{
                    //close jdbc objs & stream objects (best pratice)
                    try{
                        if(rs!=null)
              rs.close();
                        }//try
                        catch(SQLException se){
                            se.printStackTrace();
                        }

                        try{
                         if(st!=null)
              st.close();
                        }//try
                        catch(SQLException se){
                            se.printStackTrace();
                        }

                        try{
                         if(con!=null)
              con.close();
                        }//try
                        catch(SQLException se){
                            se.printStackTrace();
                        }

          try{
                        if(sc!=null)
                sc.close();
                        }//try
                        catch(Exception e){
                            e.printStackTrace();
                        }

                    }//finally
        }//main
}//class
//cmd>javac -d . SelectTest2.java
//cmd>java com.nt.jdbc.SelectTest2
```

```
G:\Worskpaces\advjava\NTAJ415\JDBC>java com.nt.jdbc.SelectTest2
enter desg1:::
clerk
enter desg2:::
manager
enter desg3:::
analyst
SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE JOB IN('CLERK','MANAGER','ANALYST') ORDER BY JOB
The Employee details are
7788  SCOTT  ANALYST  3000.0  20
7902  FORD  ANALYST  3000.0  20
7934  MILLER  CLERK  1300.0  10
7876  ADAMS  CLERK  1100.0  20
7900  JAMES  CLERK  950.0  30
7369  SMITH  CLERK  800.0  20
7698  BLAKE  MANAGER  2850.0  30
7782  CLARK  MANAGER  2450.0  10
7566  JONES  MANAGER  2975.0  20
```

Assignment :: write a JDBC App to get Student details based given 3 city names ?
Assignment :: write a JDBC App to get emp details based on the given deptno?

What is the difference b/w Editor and IDE?

Ans) Editor provides environment only to type the code.. At max to compile and execute code
where IDE (Intergrated development env..) provides the following multiple facilities
a) Development (typing the code)
b) execution
c) Debugging
d) Testing
e) Formating of code
f) Packing to jar, war and etc.. file
g) Servers integeration like Tomcat
h) Deployment of of application (Keeping java web application in to server)
i) Adding jars to CLASSPATH (Build PATH)
j) Code Coverage (checks how much code is participating for the given situation0
k) Code Commit /update to/from Code repository (GIT/SVN/ ...) (Code sharing)
l) Code suggestions or Auto Completion
m) Incremental compilation (The compilation that takes place in the Background)
and etc..

Examples of java coding editors :: notpead , wordpad , editplus, notepadd++ and etc..

Example of Java IDEs :: Eclipse , IntelliJ , STS(spring Tool suite) , netbeans, jcreator , jdeveloper and etc...
(1) (4) (2) (3)

=>Eclipse with STS plugin is more industry standard
=>Plugin is additional code /pluggable to existing s/w or s/w app to add additional features.. Eclipse is having its own pluings and also supports third party plugins

Eclipse

type :: Java IDE 4.18
version :: 2020-09 / 2020-12 /2021-03 (compitable with java8+)
Open source IDE (free IDE) 4.19
Eclipse flavousrs :: a) Eclipse SDK (we can develop onl standalone Apps)
b) Eclipse JEE (we can develop standalone Apps, web applicaitons, distributed apps and etc..)

To download Eclipse IDE :: Download as zip file
(https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2021-03/R/eclipse-jee-2021-03-R-win32-x86_64.zip&mirror_id=1093)
To install eclipse IDE :: extract zip file and lanuch eclipse.exe file ..

note:: If ur windows is 32bit then install 32 bit eclipse
If ur windows is 64bit then install 64 bit eclipse (If not avaiable then go for 32 bit)

Procedure to develop JDBC App to interact with Oracle Db s/w using Eclipse IDE

(Write a JDBC App that gives Employee details (empno,ename,job,sal) from "EMP" db table based on the initial chars of given employee name)

```
SQL> select empno,ename,job,sal from emp where ename like 'A%';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7876 ADAMS      CLERK           1100

SQL> select empno,ename,job,sal from emp where ename like 'SC%';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7788 SCOTT      ANALYST         3000
```

other related queries

```
SQL> select empno,ename,job,sal from emp where ename like '%N';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7654 MARTIN     SALESMAN        1250

SQL> select empno,ename,job,sal from emp where ename like '%N%';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7566 JONES      MANAGER         2975
      7654 MARTIN     SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7844 TURNER     SALESMAN        1500

SQL> select empno,ename,job,sal from emp where ename like '___';

no rows selected

SQL> select empno,ename,job,sal from emp where ename like '____';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7521 WARD       SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7902 FORD       ANALYST         3000

SQL> select empno,ename,job,sal from emp where length(ename)=4;

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7521 WARD       SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7902 FORD       ANALYST         3000
```

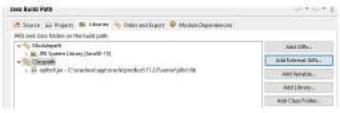
step1) Launch eclipse IDE by choosing Workspace folder (G:/workspaces/advjava/NTAJ415)
(The folder where the projects created in Eclipse IDE will be created)

step2) create JAVA Project in eclipse IDE
File menu -->new --> Project --> select java project --> name:: JDBCProj1 -->next -->next -->finish

step3) add ojdbc6.jar file to CLASSPATH /BUILDPATH of eclipse project

note:: jar files added CLASSPATH env.. variable are not visible to Projects of any IDE ... So we need to jar files to each project IDE seperately.

Right click on Project --> built path--> configure build-path --> Libraries tag ----> classpath --->
add extenal jar files ----> browse and select the jar file (ojdbc6.jar from oracle s/w installation)..



Procedure to develop JDBC App to interact with Oracle Db s/w using Eclipse IDE

(Write a JDBC App that gives Employee details (empno,ename,job,sal) from "EMP" db table based on the initial chars of given employee name)

```
SQL> select empno,ename,job,sal from emp where ename like 'AN';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7876 ADAMS      CLERK           1100

SQL> select empno,ename,job,sal from emp where ename like 'SC%';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7788 SCOTT      ANALYST         3000
```

other related queries

```
SQL> select empno,ename,job,sal from emp where ename like '%JY';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7654 MARTIN     SALESMAN        1250

SQL> select empno,ename,job,sal from emp where ename like '%JY%';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7499 ALLEN      SALESMAN        1600
      7566 JONES      MANAGER         2975
      7654 MARTIN     SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7844 TURNER     SALESMAN        1500

SQL> select empno,ename,job,sal from emp where ename like '____';

no rows selected

SQL> select empno,ename,job,sal from emp where ename like '____';

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7521 WARD       SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7902 FORD       ANALYST         3000

SQL> select empno,ename,job,sal from emp where  length(ename)=4;

     EMPNO ENAME      JOB              SAL
---------- ---------- --------- ----------
      7521 WARD       SALESMAN        1250
      7839 KING       PRESIDENT       5000
      7902 FORD       ANALYST         3000
```

step1) Launch eclipse IDE by choosing Workspace folder (D:/workspaces/edvjava/NTAJ415)

(The folder where the projects created in Eclipse IDE will be created)

step2) create JAVA Project in eclipse IDE

File menu -->new --> Project --> select java project --> name:: (DBCProj) -->next -->next -->finish

step3) add ojdbc6.jar file to CLASSPATH /BUILDPATH of eclipse project

note:: jar files added CLASSPATH env.. variable are not visible to Projects of any IDE ... So we need to jar files to each project IDE separately.

Right click on Project --> built path--> configure build-path --> Libraries tab--> classpath --> add external jar files --> browse and select the jar file (ojdbc6.jar from oracle s/w installation)..

step4) create java package in "src" folder

right click on "src" folder --> new --> package --> name :: com.nt.jdbc

step5) Develope the Java Application..

right click on "com.nt.jdbc" pkg --> new --> class --> name : SelectTest3 --> (Y)select main(-) --> ... ...

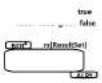
short cut keys ::

ctrl+shift+o :: To import the packages.

sysout +ctrl+space :: Gives System.out.println(-)

systrace +ctrl+space :: Gives System.out.println(-) (with message having class name and method

ctrl +shift + L :: To get All short cut keys of eclipse..




```
//SelectTest3.java
package com.nt.jdbc;

/* Java App to get Employee details based on given initial characters  Employee name */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class SelectTest3 {

	public static void main(String[] args) {
		System.out.println("SelectTest3.main()");
		Scanner scresult=null;
		Connection conresult=null;
		Statement stmtresult=null;
		ResultSet rsresult=null;
		try {
			//read inputs
			scanner Scanner(System.in);
			String initCharresult=null;
			if(sc!=null) {
			System.out.println("Enter initial characters of employee name :");
			initCharresc.next(); //gives  s
			}
			//convert  input value as required for the SQL query
			initChars=""+initCharse+"%"; //gives 'sN'

			//register JDBC driver by loading JDBC driver class
			   // Class.forName("oracle.jdbc.driver.OracleDriver");

			//establish the connection
			conDriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
			//create Statement object
			if(con!=null)
				stmcon.createStatement();
			//prepare SQL query

			// select empno,ename,job,sal  from emp where  ename like 'S%'
			String query="SELECT EMPNO,ENAME,JOB,SAL FROM EMP WHERE ENAME LIKE "+initChars;
			System.out.println(query);

			//send and execute SQL query in DB s/w
			if(st!=null)
				rsst.executeQuery(query);

			//process the ResultSet object

			if(rs!=null) {
			   boolean flag=false;
			   while(rs.next()) {
				   flag=true;
				   System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(4));
			   }//while

			   if(flag==false)
				   System.out.println("No Records  found");
			}//if

		}//try
		catch (SQLException se) {
			if(se.getErrorCode()>=900 && se.getErrorCode()<=999)
				System.out.println("invalid col names or table names or SQL keywords");
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}
		finally {
			//close jdbc objs
			try {
				if(rs!=null)
					rs.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}

			try {
				if(st!=null)
					st.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}

			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}

			try {
				if(sc!=null)
					sc.close();
			}
			catch(Exception e) {
				e.printStackTrace();
			}

		}//finally
	}//main
}//class
```

step5) Run the Application

i) Right click in .java file source code -->Run As --> Java App (or)

ii) use ctrl +F11 (or)

iii) use Run button from eclipse IDE menu bar

note1:: If SELECT SQL query gives no records ..then also JDBC ResultSet object will be created as an empty ResultSet obj

note2:: To know specific reason of raising SQL Exception we can se.getErrorCode() method ..

Those SQL Error codes DB s/w specific error codes ..

00001 :: For unique constraint violation

900 to 999 :: For SQL query related syntax problems or for invalid col names,db table names and SQL keywords

```
select * from(select ename, sal, dense_rank() over(order by sal desc) r from emp) where r=&n;
```

Sending and executing non-select SQL query in DB s/w

== INSERT,UPDATE ,DELETE SQL queries are called non-select SQL query .. when these SQL queries are executed in DB s/w they give numberic value representing no.of records that are effected.

==we use st.executeUpdate() to send and execute non-select SQL query in DB s/w.. this method returns number value (int) representing no.of records are effected.

```
    Signature :-
        public  ___int executeUpdate(String query)throws SQLException
```

Form Java 8 (working in but JDBC4.x) onwards we got one more method to execute non-select SQL queries having return type long and the method is executeLarge(Update) ;

```
  Signature :-
        public     long  executeLarge(Update)(String query)throws SQLException
```

==Latest version of JDBC is 4.x even Java 18 is dealing JDBC 4.x version
== JDBC ,ODCE technologies are part of JSE module nothing JDK s/w .. So there is no need of arranging JDBC API seperately..

==the non-select SQL qulery that is sent to oracle DB s/w from SQL prompt /SQL Plus (readymade clients of oracle DB s/w) will execute in DB s/w by disabling autocommit mode in DB s/w.. So we need to issue "commit" command explicitly inorder to save the changes perminently.

```
    SQL> delete from student where  sadd='hyd';
    1 row deleted.
    SQL> commit;  ( execution)
```

== The non-select SQL query that is sent to DB s/w from Java App (with JDBC code) executes in DB s/w by enabling auto commit mode on DB s/w.. So there is no need of sending "commit" command to DB s/w from Java app seperately..

Write a JDBC App to delete student records based given city name (sadd)

shortcut :-To Format ,To perform indentation of the selected code then go for ctrl+I

conc: Java Apps with JDBC can not read uncommitted data from DB s/w...

```
//student.java
package com.nt.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class DeleteTest {

    public static void main(String[] args) {
        Scanner sc=null;
        Connection conn=null;
        Statement stmt=null;
        try {
            //read inputs
            sc=new Scanner(System.in);
            String city=null;
            if(sc!=null) {
                System.out.println("Enter  student address (city name) :");
                city=sc.next(); //gives  hyd
            }
            else**"+e.toString()); //gives "null"

            //register  JDBC driver by loading  JDBC driver class
        // Class.forName("oracle.jdbc.driver.OracleDriver");

            //establish the connection
            conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
            //create Statement object
            if(conn!=null)
                stmt=conn.createStatement();
            //prepare SQL query
            //delete from student where  sadd='hyd'
            String query="DELETE FROM STUDENT WHERE  SADD='"+city;
            System.out.println(query);

            //send and execute SQL query in DB s/w
            int count=0;
            if(stmt!=null)
                count=stmt.executeUpdate(query);

            //process the result
            if(count==0)
                System.out.println("No records found to delete");
            else
                System.out.println("no.of records that are effected :"+count);
        }//try
        catch(SQLException se) {
            if(se.getErrorCode()>=900 && se.getErrorCode()<=999)
                System.out.println("Invalid col names or table names or SQL keywords");
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            //close JDBC objs

            try {
                if(stmt!=null)
                    stmt.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }

            try {
                if(conn!=null)
                    conn.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }

            try {
                if(sc!=null)
                    sc.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }

        }//finally
    }//main
}//class
```

Assignment :- Write JDBC App to delete student db table records based on the given student number ?

Write a JDBC to modify student name, avg & sadd based on the given student number(snio)

(This test completion needs UP5470 SQL query)

```
SQL> update  student set sname='anil' ,sadd='delhi' ,avg=91.55 where sno=1005;

1 row updated.

SQL> commit;

Commit complete.

SQL> update  student set sname='anil' ,sadd='delhi' ,avg=91.55 where sno=1005;

0 rows updated.
```

example app

```
//UpdateTest.java
package com.nt.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class UpdateTest {

    public static void main(String[] args) {
        Scanner sc=null;
        Connection conn=null;
        Statement stmt=null;
        try {
            //read inputs
            sc=new Scanner(System.in);
            String newCity=null,newName=null;
            float newAvg=0.0f;
            int sno=0;
            if(sc!=null) {
                System.out.println("Enter new name for student :");
                newName=sc.next(line); //gives anil sen
                System.out.println("Enter new address for student :");
                newCity=sc.next(line); //gives rani soundal
                System.out.println("Enter new avg for student :");
                newAvg=sc.nextFloat(); //gives 99.99
                System.out.println("Enter sno of student :");
                sno=sc.nextInt(); //gives 1003
            }
            //Convert input values as required for the SQL query
            newName="'"+newName+"'"; //gives 'anil sen'
            newCity="'"+newCity+"'"; //gives 'rani soundal'

            //register  JDBC driver by loading  JDBC driver class
        // Class.forName("oracle.jdbc.driver.OracleDriver");

            //establish the connection
            conn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
            //create Statement object
            if(conn!=null)
                stmt=conn.createStatement();
            //prepare SQL query
              //Update  student set sname='anil sen' ,sadd='rani soundal' ,avg=91.55 where sno=1003;
              String query="update  student set sname="+newName+" ,sadd="+newCity+" ,avg="+newAvg+" where sno="+sno;
              System.out.println(query);

            //send and execute SQL query in DB s/w
            int count=0;
            if(stmt!=null)
                count=stmt.executeUpdate(query);

            //process the result
            if(count==0)
                System.out.println("No records found for updation");
            else
                System.out.println("no.of records that are effected :"+count);
        }//try
        catch(SQLException se) {
            if(se.getErrorCode()>=900 && se.getErrorCode()<=999)
                System.out.println("Invalid col names or table names or SQL keywords");
            else if(se.getErrorCode()==12899)
                System.out.println("Do not insert more than col size data to sname,sadd cols");

            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            //close JDBC objs

            try {
                if(stmt!=null)
                    stmt.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }

            try {
                if(conn!=null)
                    conn.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }

            try {
                if(sc!=null)
                    sc.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }

        }//finally
    }//main
}//class
```

==st.executeUpdate() (and st.executeLarge(Update)() method can perform single row non select operations and bulk non-select operations based on the conditions we pass in those SQL queries

Assignment :- a) Write a JDBC App to hike a mp sales by given percentage for the employees whose salary is in the given range (start range to end range) (total 3 inputs)

b) Write a JDBC App to add given percentage of marks to existing avg based on the given 4 city names as the addresses for student. (total 4 inputs)

Can we use single method call on JDBC Statement obj to send and execute
both SELECT and NON-SELECT Query in Db s/w (But One SQL query at a time)

Ans) possible ... by execute(-) invocation on JDBC statement object .. But not recomanded to use..
Now
prefer executeUpdate(-) to send and execute SELECT SQL query to Db s/w and prefer
executeQuery(-) to send and execute select SQL query.

signature of execute(-) method
==========
public boolean execute(String query) throws SQLException

=>If execute(-) is sending and executing SELECT SQL Query then this metod
returns true and we need to call st.getResultSet() method seperately
to get ResultSet object

=>If execute(-) is sending and executing Non-SELECT SQL Query then this
metod returns false and we need to call st.getUpdateCount() method
seperately to get numeric value representing no.of records that are
effected.

```
boolean b=st.execute(query);
if(b==true){
  S.o.p("SELECT SQL query is executeed");
  ResultSet rs=st.getResultSet();
  //process the ResultSet
    while(rs.next()){
      ....
      ....
    }
}
else {
  S.o.p("NON-SElect SQL query is executed");
  int count=st.getUpdateCount();
  S.o.p("no.of records that are effected are :"+count);
}
```

note:- while dealing with execute(-) mnethod we need to involve total two methods
1) execute(-) to send and exectue SQL query to Db s/w
2) getResultSet()/(for SELECT SQL)/getUpdateCount()(for NON-SELECT SQ)
to gather results of the executed SQL query as show above.. So
execute(-) is not recomanded to use .. prefer using executeUpdate(-) method for
NON-SELECT SQL query and executeQuery(-) method for SELECT SQL query execution.

Diffrunt ways of performing soft coding (gathering inputs from Endoser) in Java apps
==========
a) cmd line args
b) System properties
c) Using IO streams like BufferedReader
d) Using java.util.Scanner (S)
e)Using java.util.Console
f) using properties file (S)
g) using xml file (S)
h) using JSON file (maintains info as key,value pairs as alternate to xml file)
i) AWT/Swing Form comps
j) HTML Form comps (S)

AWT :: Abstact Window Toolkit
JSON :: Java script Object Notation format
(alternate no xml file and better
compare to properties file and other techniques.

and etc..

//SelectNonSelectTest.java
package com.nt.jdbc;

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class SelectNonSelectTest {

    public static void main(String[] args) {
        //read inputs
        Connection con=null;
        Scanner sc=null;
        Statement st=null;
        ResultSet rs=null;
      try {
          sc=new Scanner(System.in);
          int cp=0;
          String query=null;
          if(sc!=null) {
              System.out.println("Enter SQL query(select or non-select :");
              query=sc.next().trim();
          }//read inputs

          //Load JDBC driver class
          //Class.forName("oracle.jdbc.driver.OracleDriver");
          //Establish the connection

          //gather the results from Db s/w and use them
          con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

          //create JDBC Statement obj
          if(con!=null)
              st=con.createStatement();
          //send and execute SQL query in Db s/w
          if(st!=null ) {
              boolean flag=st.execute(query);
              System.out.println(flag);
              if(flag==true) {
                  System.out.println("SELECT SQL Query executed");
                  //gather and process the ResultSet
                  rs=st.getResultSet();
                  //process the ResultSet obj
                  if(rs!=null) {
                      while(rs.next()) {
                          System.out.println(rs.getString(1)+" "+rs.getString(2)+"  "+rs.getString(3));
                      }//while
                  }//if

              }//if
              else {
                  System.out.println("Non-SELECT SQL Query executed");
                  //gather result
                  int count=st.getUpdateCount(); // long count=st.getLargeUpdateCount();
                  System.out.println("no.of records that are effected.:"+count);
              }//else
          }//if
      }//try
      catch(SQLException se) {
          se.printStackTrace();
          //we can work with error codes
      }
      catch(Exception e){
          e.printStackTrace();
      }
      finally {
          //close JDBC objs
          try {
              if(st!=null)
                  st.close();
          }
          catch(SQLException se) {
              se.printStackTrace();
          }
          try {
              if(rs!=null)
                  rs.close();
          }
          catch(SQLException se) {
              se.printStackTrace();
          }
          try {
              if(con!=null)
                  con.close();
          }
          catch(SQLException se) {
              se.printStackTrace();
          }
          try {
              if(sc!=null)
                  sc.close();
          }
          catch(Exception e) {
              e.printStackTrace();
          }
      }//finally
  }//main
}//class
```

Execution1::
enter SQL query(select or non-select ::
update dept set Loc='S2' where deptno=50
false
Non-SELECT SQL Query executed
no.of records that are effected::1

execution2::
enter SQL query(select or non-select ::
select * from student
true
SELECT SQL Query executed
567 ramesh hyd
167 ramana mumbai
1001 karan navi hyd
556 karan hyd
1010 ramesh vizag

Limitations of this Application
==========
(a) This app just good as POC (proof of concept) .. Real implementation is not possible
(b) Collecting SQL query from non-technical endoser is pratical not possible
(c) in real pratices select SQL query will not be changed all of sudden to non-select Query
So there is not this kind of App development

(d) while dealing with execute(-) mnethod we need to involve total two methods
1) execute(-) to send and exectue SQL query to Db s/w
2) getResultSet()/(for SELECT SQL)/getUpdateCount()(for NON-SELECT SQ)

(e) It is not industry standard approch to develop jdbc code..
and etc..

Assignment:: Can we perform DDL operations using JDBC . If so what
is the method we should call on statement obj to send and
execute SQL query.
drop
DDL operations:: create table, alter table and delete table

Sending and executing DDL Queries in DB s/w

=>DDL queries [Data Definition Language Queries] :: create table, alter table ,drop table and etc... called DDL queries ... As Java app developer we treat them as Non-select Queries and use use execute(update()) to send and execute these queries in Db s/w.

=> executing DDL Queries from the Front end app (like Java App) is always for knowlege or practice sake ... bacoz every company maintains seperate DB Designing Team to design the Db tables for the project having relationships and also by applying 3 forms of normalization..

A typical Java project contains

If we can do all these things then u will be called as fullstack java developer

a) Java developers
b) testers
c) SQL Developers /DB team
d) UI developers
e) Build and Deployment teams (DevOps Engineers) | (Only in large scale projects that uses Cloand env... like AWS/Azure / Google Cloud /....)
f) Cloud Engineers

CreateTableTest.java

```
package com.nit.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class CreateTableTest {

	public static void main(String[] args) {
		Connection con=null;
		Statement stmt=null;
		try {
			//Load JDBC driver class
		//Class.forName("oracle.jdbc.driver.OracleDriver");
		//establish the connection

		//gather the results from Db s/w and use there
		con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

		//create JDBC Statement obj
		if(con!=null)
			stmt=con.createStatement();
		//prepare SQL query
		 //create table temp_student(sno number(4) primary key ,sname varchar2(15));
		 String query="CREATE TABLE  TEMP_STUDENT(SNO NUMBER(5) PRIMARY KEY ,SNAME VARCHAR2(15))";
		 System.out.println(query);
		 //send and execute SQL query to Db s/w
		 int count=0;
		 if(stmt!=null)
			 count=stmt.executeUpdate(query);
		 System.out.println(" count: "+count);
		 //process the result
		 if(count==0)
			 System.out.println("Db table is  created");
		 else
			 System.out.println("Db table is not  created");

		}//try
		catch(SQLException se) {
			se.printStackTrace();
			//we can work with error codes
			if(se.getErrorCode()==955)
				System.out.println("DB table is already created");
		}
		catch(Exception e){
			e.printStackTrace();
		}
		finally {
			//close  JDBC objs

			try {
				if(st!=null)
					st.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
		}//finally
	}//main
}//class
```

output:

```
CREATE TABLE  TEMP_STUDENT(SNO NUMBER(5) PRIMARY KEY
,SNAME VARCHAR2(15))
count:0
Db table is  created
```

note: some JDBC drivers and some DB s/w do not give correct numbers value while dealing DDL queries as non-select SQL queries

For example while talking to oracle Db s/w using oracle thin JDBC driver the execute(Update()) that is sending and executing create table DDL query returns 0 for successful creation of Db table and throws exception of db table creation is failed. In some Db s/ws and JDBC drivers this return value may "1" for successful creation of Db table

DropTableTest.java

```
//DropTableTest.java
package com.nit.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class DropTableTest {

	public static void main(String[] args) {
		Connection con=null;
		Statement stmt=null;
		try {
			//Load JDBC driver class
		//Class.forName("oracle.jdbc.driver.OracleDriver");
		//establish the connection

		//gather the results from Db s/w and use there
		con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

		//create JDBC Statement obj
		if(con!=null)
			stmt=con.createStatement();
		//prepare SQL query
		 //drop table temp_student
		 String query="DROP  TABLE  TEMP_STUDENT";
		 System.out.println(query);

		 //send and execute SQL query to Db s/w
		 int count=0;
		 if(stmt!=null)
			 count=stmt.executeUpdate(query);
		 System.out.println(" count: "+count);
		 //process the result
		 if(count==0)
			 System.out.println("Db table is dropped");
		 else
			 System.out.println("Db table is not  created");

		}//try
		catch(SQLException se) {
			se.printStackTrace();
			//we can work with error codes
			if(se.getErrorCode()==955)
				System.out.println("DB table is already created");
		}
		catch(Exception e){
			e.printStackTrace();
		}
		finally {
			//close  JDBC objs

			try {
				if(st!=null)
					st.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
		}//finally
	}//main
}//class
```

Assignment :: Develop a app perform "alter table operation".

Login App development

=>it is the process of performing authentication nothing but verifying given username, password details against the registered user details of Db table.

step1) Take a Db table having registered user details (including usernames and password)

```
SQL> create table IRCTC_TAB(uname varchar2(20),pwd varchar2(20),DOB date, location varchar2(20));
Table created.
```

step2) Insert records into db table representing the registered customer details..

```
SQL> insert into IRCTC_TAB values('raja','ravi','20-OCT-90','hyd');

1 row created.

SQL> insert into IRCTC_TAB values('ravi','hyd','21-OCT-1995','hyd');

1 row created.

SQL> insert into IRCTC_TAB values('king','kingdom','22-OCT-1970','delhi');

1 row created.

SQL> insert into IRCTC_TAB values('jani','begum','22-OCT-1987','mumbai');

1 row created.

SQL> insert into IRCTC_TAB values('anil babu','rao rao','21-NOV-1979','hyd');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from IRCTC_TAB;

UNAME(pk)    PWD          DOB       LOCATION
-------------------------------------------------
raja         ravi         20-OCT-90 hyd
ravi         hyd          21-OCT-95 hyd
king         kingdom      22-OCT-70 delhi
jani         begum        22-OCT-87 mumbai
anil babu    rao rao      21-NOV-79 hyd
```

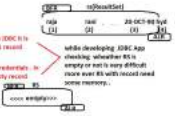
step3) Decide the SQL query for authentication as part of Login activity...

```
SQL> select * from IRCTC_TAB where uname='raja' and pwd='ravi';
UNAME        PWD          DOB       LOCATION
-------------------------------------------------
raja         ravi         20-OCT-90 hyd
                  (valid credentials)

SQL> select * from IRCTC_TAB where uname='raja' and pwd='ravi1';
no rows selected
                  (invalid credentials)
```

For valid credentials ... in JDBC it is gaining rs table RS with 1 record

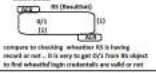
For invalid credentials .. in RS with empty record

while developing JDBC App checking whether RS is empty or not is very difficult more over RS with record need some memory..




```
SQL> select count(*) from IRCTC_TAB where uname='raja' and pwd='ravi';
  COUNT(*)
----------
         1
                  (valid credentials)
SQL> select count(*) from IRCTC_TAB where uname='raja' and pwd='ravi1';
  COUNT(*)
----------
         0
                  (invalid credentials)
```

Given RS having 0/1 (1 record with 1 value) as shown below



compare to checking whether RS is having record or not .. it is very to get 0/1 from RS object to find whaatthd login credentails are valid or not

---

step4) develope the Applications...

Login App development

==It is the process of performing authentication nothing but verifying given username, password details against the registered user details of Db table.

step1) Take a Db table having registered user details (including usernames and password)

```
SQL> create table IRCTC_TAB(uname varchar2(20),pwd varchar2(20),DOB date, location varchar2(20));
Table created.
```

step2) Insert records into db table representing the registered customer details..

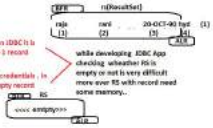
```
SQL> insert into IRCTC_TAB values('raja','rani',20-OCT-90','hyd');

1 row created.

SQL> insert into IRCTC_TAB values('rani','hyd',21-OCT-1995','hyd');

1 row created.

SQL> insert into IRCTC_TAB values('king','kingdom','22-OCT-1970','delhi');

1 row created.

SQL> insert into IRCTC_TAB values('jani','begum','22-OCT-1987','mumbai');

1 row created.

SQL> insert into IRCTC_TAB values('anil babu','rao rao','21-NOV-1973','hyd');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from IRCTC_TAB;

UNAME (PK)      PWD             DOB      LOCATION
--------------- --------------- -------- ---------------
raja            rani            20-OCT-90 hyd
rani            hyd             21-OCT-95 hyd
king            kingdom         22-OCT-70 delhi
jani            begum           22-OCT-87 mumbai
anil babu       rao rao         21-NOV-73 hyd
```

step3) Decide the SQL query for authentication as part of Login activity..

```
SQL> select * from IRCTC_TAB where uname='raja' and pwd='rani';
UNAME           PWD             DOB      LOCATION
--------------- --------------- -------- ---------------
raja            rani            20-OCT-90 hyd
                    (valid credentials)

SQL> select * from IRCTC_TAB where uname='raja' and pwd='rani1';
no rows selected
                    (invalid credentials)
```

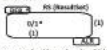
For valid credentials ... in JDBC it is going to table RS with 1 record

For invalid credentials , in RS with empty record




```
SQL> select count(*) from IRCTC_TAB where uname='raja' and pwd='rani';
 COUNT(*)
---------
        1     (valid credentials)
SQL> select count(*) from IRCTC_TAB where uname='raja' and pwd='rani1';
 COUNT(*)
---------
        0     (invalid crendetials)
```

Gives RS having 0/1 (1 record with 1 value) as shown below



compare to checking wheather RS is having record or not .. it is very to get 0/1 from RS object to find wheathd login credentials are valid or not

---

step4) develope the Application...

```
package com.nt.jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class LoginApp {

	public static void main(String[] args) {
		Scanner scread=null;
		Connection con=null;
		Statement st=null;
		ResultSet rs=null;
		try {
			//read inputs
			scr=new Scanner(System.in);
			String user=null,pass=null;
			if(sc!=null) {
				System.out.println("enter Login username:-");
				user=sc.next(line); //gives raja rao
				System.out.println("enter Login password:-");
				pass=sc.next(line); //gives raa raa
			}//if
			//convert input values as required for the SQL query
			user="'"+user+"'"; //gives 'raja rao'
			pass="'"+pass+"'"; //gives 'raa raa'
			//Load JDBC driver class (optional)
			Class.forName("oracle.jdbc.driver.OracleDriver");
			//establish the connection
			con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
			//create JDBC statement object
			if(con!=null)
				st=con.createStatement();
			//prepare SQL query
				//select count(*) from IRCTC_TAB WHERE uname='raja' and pwd='rani';
			String query="SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME="+user+" AND PWD="+pass;
			System.out.println(query);

			//send and execute the SQL query in Db s/w
			if(st!=null)
				rs=st.executeQuery(query);

			//process the ResultSet object
			if(rs!=null) {
				rs.next(); //moves the curser to first record from BFR
				int count=rs.getInt(1); // get first col value of that first record
				//process the result
				if(count==0)
					System.out.println("INVALID CREDENTIALS");
				else
					System.out.println("VALID CRIDENTIALS");
			}//if
			}//try
		catch(SQLException se) {
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}
		finally {
			//close jdbc objs
			try {
				if(rs!=null)
					rs.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(st!=null)
					st.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}

			try {
				if(sc!=null)
					sc.close();
			}
			catch(Exception e) {
				e.printStackTrace();
			}
		}//finally

	}//main
}//class
```

```
enter Login username::
anil babu
enter Login password::
rao rao
SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME='anil babu' AND PWD='rao rao'
VALID CREDENTIALS
```

```
enter Login username::
anil babu
enter Login password::
raa ra
SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME='anil babu' AND PWD='raa ra'
INVALID CREDENTIALS
```

==While developing LoginApp if want to avoid echoing of password on console monitor then we need to use " Console.readPassword() method" as shown below

```
Console con=System.console();  //java.util.Console does not work Eclipse IDE
	String username,password;
	if(con!=null) {
		System.out.println("enter Login username:-");
		user=scrs.readLine(); //gives raja rao
		System.out.println("enter Login password:-");
		pass=new String(con.readPassword()); //gives raa raa
	}//if
```

Eclipse IDE is not still supporting Console class.. So use Scanner support in Eclipse IDE.

cmd

Qi) What is SQL Injection Problem? x of the App

Ans) passing special SQL instructions(like -- symbol) to the SQL query along with the input value and changing the behaviour of SQL query and the underlying app is called SQL injection problem.

=> Hackers ,sackers use this technique to hack the account with out knowing usernames or passwords or both.

usecase1::

```
	enter Login username::
	raja' --
	enter Login password::
	hyd                (wrong password)
	SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME='raja' --' AND PWD='hyd'
	VALID CREDENTIALS  (SQLinjection problem)
```

usecase2:

```
enter Login username::
xyz' or 1=1 --        (wrong username)
enter Login password::
hyd                (wrong password)
SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME='xyz' or 1=1 --' AND PWD='hyd'
VALID CREDENTIALS   (SQL Injection Problem)
```

JDBC PreparedStatement object

java.sql.Statement(I)
↑ extends
java.sql.PreparedStatement(I)
↑ extends
java.sql.CallableStatement(I)

3 types of JDBC Statement objects
a) Simple Statement obj :: It is the implementation class obj of java.sql.Statement(I).
b) PreparedStatement obj :: It is the implementation class obj of java.sql.PreparedStatement(I)
c) CallableStatement obj :: It is the implementation class obj of java.sql.CallableStatement(I)

Limitation of Simple Statement obj

=>Preparing SQL query by involving varlables is very complex
=>We need to convert input values as required for SQL query
=>Does not allow to take parameters/place holders(?) in the SQL Query whose values can be set later.
[This indicates query must be prepared having input values and we can not place ? in the SQL query repreposenting values and allowing to set values later]
=> We insert Date values using SimpleStatement obj.. But the data values must be given in the pattern that the underlying Db s/w supports.
oracle date pattern :: dd-MMM-yy eg:: 10-OCT-90
mysql date pattern :: yyyy-MM-dd eg:: 1990-10-10

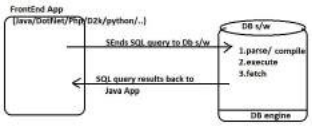
=>Not suitable to execute same SQL query for multiple times either with same values or with different values
=> May raise SQL Injection Problem
=>Not suitable for inserting Large objects(files) to db table cols
=>Working with Simple Statement obj is not a Industry standard.
and etc..

=>If we give SQL Query to Db s/w from any front end app, the DB engine of DB s/w performs the following 3 operations

1.parse/compile
=> splits the query in to tokens and checks the syntax of the SQL Query

2.execute
=> executes the parsed/compiled SQL query and generates results

3.fetch
=>Gathers the generated results of SQL query and sends those results back to Frontend App



If we use Simple Statement obj to send and execute same SQL query for multiple times:::

a) same SQL query goes to Db s/w for multiple times either with same inputs or different inputs
b) same SQL query will be parsed/compiled in s/w for multiple times either with same inputs or different inputs

These operations (a),(b) on same SQL query for multiple times unneccesary ..but we can not avoid while working with Simple statement obj So the SimpleStatement is not suitable for executing same SQL query for multiple times

c)same SQL query will be executed in s/w for multiple times either with same inputs or different inputs
d)same SQL query outputs/results will be fetached out from DB s/w for multiple times
e)same SQL query outputs/results will be sent to Front End App from Db s/w for multiples times

These operations (c),(d),(e) on the same SQL qery for multiple times is required..

Indian railway ticket reservation App (Booking 1,00,000 lac tickets in day) (Using Simple statement obj)

(a) sends same INSERT SQL query to Db s/w from Front End with diff input values for 1,00,000 times :: 1,00,000 * 0.1 sec = 10,000 secs (Not required for multiple times)
(b) Db s/w parses/compiles the same INSERT SQL query with differnet input values for 1,00,000 times :: 1,00,000 * 0.1 sec = 10,000 secs (Not required for multiple times)
(c) Db s/w executes the same INSERT SQL query with differnet input values for 1,00,000 times :: 1,00,000 * 0.1 sec = 10,000 secs (required for multiple times)
(d) Db s/w fetches the same INSERT SQL query results for 1,00,000 times :: 1,00,000 * 0.1 sec = 10,000 secs (required for multiple times)
(e) Db s/w sends the same INSERT SQL query results back to Java App for 1,00,000 times :: 1,00,000 * 0.1 sec = 10,000 secs (required for multiple times)

50,000 secs

=>20,000 secs wasted in a day

To the solve the problems , we need to PreparedStatement obj that deals with pre-compiled SQL query.

What is pre-compiled SQL query

Ans) The SQL query that goes to Db s/w and becomes parsed/compiled SQL query irrespective of wheather it will be executed or not is called pre-compiled SQL query.
=> Once pre-compiled SQL query is ready , It will be represented JDBC prepared statement obj and when this used to send and execute pre-compiled SQL query the DB s/w will not parse/compile the SQL query again becoz query is already compiled(pre-compiled)

Indian railway ticket reservation App (Booking 1,00,000 lac tickets in day) (Using PrpearedStatement obj)

a) SQL INSERT query with params(?) goes to Db s/w to make the SQL query as the pre-compiled SQL query and gives that pre-comiled querry back to Java App to represent through PreparedStaement obj for 1 time :: 1 * 0.1 sec : 0.1 sec

(b) Java App uses PreparedStatement obj to set input values to query params and to send pre-compiled SQL query to Db s/w for 1,00,000 times :: 1,00,000 * 0.1sec = 10,000 secs

(c) Pre-compiled SQL query with input values will be executed for for 1,00,000 times :: 1,00,000 * 0.1 sec =10,000 secs

(d) SQL query output will be fetechted our for for 1,00,000 times :: 1,00,000 * 0.1 sec =10,000 secs

(e) SQL query results will be sent back to Front End App(Java App) for 1,00,000 time :: 1,00,000 * 0.1 sec =10,000 secs

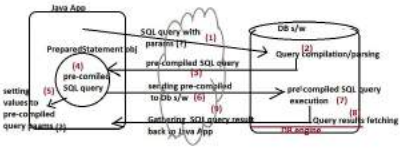
40,000.01 secs

Here wastage of time is minimized to 10,000 secs in a day

note:: once DB s/w make the SQL query as pre-compiled , It comes to back Java App and will be stored in PreParedStatement obj and using PreparedStatement obj we can do following operations on the pre-compiled SQL query

a) set input values to query params
b) sending and executing query for multiple timnes
c) fetching out query execution results for multiple times
d) gathering query execution results back to Java app for multiple times

PreSatement obj is like remote to the TV.



note:: JDBC PreparedStatement is good to execute Same SQL query for multiple times either with same inputs or with different inputs.

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

step1) prepare SQL query having parameters/place holders(?) representing the input values
eg:
String query="INSERT INTO STUDENT VALUES(?,?,?,?)";
1 2 3 4

step2) send the SQL query to Db s/w , make it as pre-compiled SQL query and get JDBC PreperedStatement object representing/holding that pre-compiled SQL query
eg:
PreparedStatement ps=con.prepareStatement(query);
=>This method sends given SQL query to Db s/w and makes the SQL query as pre-compiled SQL query in Db s/w and also returns PreparedStatement obj back to Java App representing the pre-compiled SQL query.
note:: ps represents JDBC PreparedStatement obj having pre-compiled SQL query.
PreparedStatement obj means it is the object of underlying JDBC driver s/w supplied Java class that implements java.sql.PreparedStatement(I).
ps
having INSERT Query
as pre-compiled SQL query

step3) set values to the query params of pre-compiled SQL query using ps.setXxx(-,-)
(?)
ps.setInt(1,1001);
ps.setString(2,"raja");
ps.setString(3,"hyd");
ps.setFloat(4,99.77f);
param index
param value

step4)send and execute the pre-compiled SQL query to Db s/w
int result=ps.executeUpdate();

step5) process the result
if(result==0)
S.o.p("record is not inserted");
else
S.o.p("record is inserted");

note:: To execute the same SQL qery for multiple times with different inputs we need repeate step3,4,5 for multiple times..

step6) close JDBC objs
ps.close();
con.close();

---

note:: we can make both select and non-select SQL query as pre-compiled SQL query using the supported PreparedStatement obj

conclusion ::
If u want to execute SQL query only for one time that to with out any input values then go for simpleStatement .. In remaining all situations we need to use PreparedStatement object.

Q) Write a JDBC App to insert "n" student details to "STUDENT" db table by collecting details from endusers

=>Here we need to execute same INSERT SQL query for multiple times with different input values So prefer using JdbcPreparedStatement objet that deals with pre-compiled SQL query.

=>while working pre-compiled SQL queries it recomanded to SQL query at top of the java class definitation as constant variable (private/public static final variable) value.. This process allows to modify all SQL queries of the App by going to single place when ever needed.

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

PsInsertTest.java

```
package com.rit.jdbc1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

public class PsInsertTest {
  private static final String STUDENT_INSERT_QUERY="INSERT INTO STUDENT VALUES(?,?,?,?)";
    public static void main(String[] args) {
        Scanner sc=null;
        Connection con=null;
        PreparedStatement ps=null;
         try {
              //read inputs
              sc=new Scanner(System.in);
              int count=0;
              if(sc!=null) {
                   System.out.println("enter students count:");
                   count=sc.nextInt();
              }
              //register JDBC driver
               //Class.forName("com.mysql.cj.jdbc.Driver");

              //establish the connection
              con=DriverManager.getConnection("jdbc:mysql:///NTA0435DB","root","root");
             // con=DriverManager.getConnection("jdbc:mysql://localhost:3306/NTA0435DB","root","root");

              //create PreparedStatement object having pre-compiled SQL query
              if(con!=null)
                ps=con.prepareStatement(STUDENT_INSERT_QUERY);
               //read input values from enduser ,set them to query param values and execute the pre-compiled
               //SQL query for multiple times
               if(ps!=null && sc!=null) {
                    for(int i=1;i<=count;++i) {
                         //read each student input values
                         System.out.println("enter "+i+" student details");
                         System.out.println("enter student number:");
                         int no=sc.nextInt();
                         System.out.println("enter student name:");
                         String name=sc.next();
                         System.out.println("enter student address:");
                         String addrs=sc.next();
                         System.out.println("enter student avg:");
                         float avg=sc.nextFloat();
                         //set each student details as pre-compiled SQL query params
                         ps.setInt(1, no); ps.setString(2, name); ps.setString(3, addrs); ps.setFloat(4, avg);
                         //execute pre-compiled SQL query each time
                         int result=ps.executeUpdate();
                         //process execution result of pre-compiled SQL query
                         if(result==0)
                              System.out.println(i+" student details not inserted");
                         else
                              System.out.println(i+" student details are inserted");
                    }//for
               }//if

         }//try
         catch(SQLException se) {
              se.printStackTrace();
         }
         catch(Exception e) {
              e.printStackTrace();
         }
         finally {
              //close jdbc objs
              try {
                   if(ps!=null)
                        ps.close();
              }
              catch(SQLException se) {
                   se.printStackTrace();
              }
              try {
                   if(con!=null)
                        con.close();
              }
              catch(SQLException se) {
                   se.printStackTrace();
              }
              try {
                   if(sc!=null)
                        sc.close();
              }
              catch(Exception e) {
                   e.printStackTrace();
              }
         }//finally

    }//main
}//class
```

Write a JDBC App to insert multiple customer details to Customer Db table?

Login App with PreparedStatement obj
=====================================

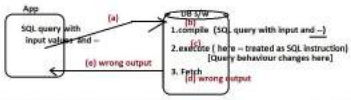
```
package com.nt.jdbc1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class PSLoginApp {
  private static final String  LOGIN_QUERY="SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME=? AND PWD=?";
      public static void main(String[] args) {
            Scanner sc=null;
            Connection con=null;
            PreparedStatement ps=null;
            ResultSet rs=null;
            try {
                  //read inputs
                  sc=new Scanner(System.in);
                  String user=null,pass=null;

                  if(sc!=null) {
                        System.out.println("enter Login username::");
                        user=sc.nextLine();  //gives raja rao' --
                        System.out.println("enter Login password::");
                        pass=sc.nextLine();  //gives rao rao
                  }//if

                  //Load  JDBc driver class  (optional)
                  Class.forName("oracle.jdbc.driver.OracleDriver");
                  //establish the connection
                  con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                  //create  JDBC PreparedStatement object
                  if(con!=null)
                        ps=con.prepareStatement(LOGIN_QUERY);

                  //set values to  the params of pre-compiled SQL query
                  if(ps!=null) {
                        ps.setString(1,user);
                        ps.setString(2, pass);
                  }

                   //send and execute the SQL query in Db s/w
                   if(ps!=null)
                        rs=ps.executeQuery();

                  //process the ResultSet object
                  if(rs!=null) {
                        rs.next();  //moves the cursor to  first record from BFR

                        int count=rs.getInt(1);  // get first col value of that first record

                        //process the result
                        if(count==0)
                              System.out.println("INVALID CREDENTIALS");
                        else
                              System.out.println("VALID CREDENTIALS");
                  }//if
                  }//try
               catch(SQLException se) {
                     se.printStackTrace();
               }
               catch(Exception e) {
                     e.printStackTrace();
               }
               finally {
                     //close jdbc objs
                     try {
                           if(rs!=null)
                                 rs.close();
                     }
                     catch(SQLException se) {
                           se.printStackTrace();
                     }
                     try {
                           if(ps!=null)
                                 ps.close();
                     }
                     catch(SQLException se) {
                           se.printStackTrace();
                     }
                     try {
                           if(con!=null)
                                 con.close();
                     }
                     catch(SQLException se) {
                           se.printStackTrace();
                     }

                     try {
                           if(sc!=null)
                                 sc.close();
                     }
                     catch(Exception e) {
                           e.printStackTrace();
                     }
               }//finally

      }//main
}//class
```

=>PreparedStatement obj contains pre-compiled and natually it does not have SQL Injection problem..

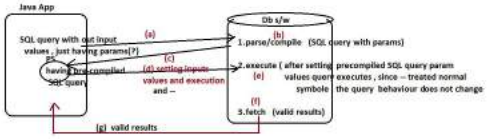
usecase1:	usecase2:
enter Login username::	enter Login username::
xyz' or 1=1 -- (wrong username)	raja' -- (correct username)
enter Login password::	enter Login password::
hyd (wrong password)	hyd (wrong password)
INVALID CREDENTIALS	INVALID CREDENTIALS (No SQL Injection problem)
(No SQL Injection problem)	

Q) Why simple Statement obj having SQL injection problem and why PreparedStatement obj is not having SQL injection problem?

Ans) Simple Statement obj makes DB s/w to compile the SQL query directly having input values..So the special SQL instructions that are given along with the input values (like -- symbol) participates in SQL query compilation, So when the compiled SQL Query is given for execution the SQL instructions change the behaviour SQL query and thus changes the application behaviour



Ans) While Working with Preparedstatement obj ,the Db s/w makes the SQL Query as pre-compiled SQL query with out input vlaues i.e input values and special SQL instructions like -- that comes along with input values do not participate in SQL query compilation. More over input values are set to pre-compiled SQL query directly just before execution ,So the input vlaues and and sql instructions like -- are directly participating SQL query execution.. due to this the SQL instructions like -- will be treated as normal symbols and they does not change SQL query and application behaviour .. This indicates there is no SQL Injection Problem.



Advantages of JDBC PreparedStatement obj
==

a) Supports to hold pre-compiled SQL queries
b) No need of converting input values as the required for the SQL query
c) We can take SQL queries with parameters/placeholders
d) Input values can be set SQL query parameters directly with out converting them
e) No SQL Injection problem
f) Suitable to execute same SQL query for multiple times either same inputs or different inputs
g) Date values can be inserted to db table cols by collecting them fromuser in diffrent patterns
h) Overall improves the performence
i) Suitable for inserting Large objects (files)
j) It is industry standard JDBC Statement obj in JDBC Apps devleopment.

and etc..

Generating PK column Value dynamically (Logical keys)

While doing insert opertions /registration operations in any Db table through FrontEnd App , we do not ask enduser to supply PK column column value .. it will be generated by App or DB s/w dynamically
eg:: sno for Student , eno for Employee, registration no for counselling table , billNo for Payments db table and etc..

=>The column in Db table that holds unique values and whose values can be used to indentify and access the records is called Candidate key column

Person_info (db table)

pid (pk) (n) -->candidate key column
pname (vc2)
paddrs (vc2)
aadharNo (n)
passportNo(n)
panNo (n)
voterId (n)
drivingLicense (n)
mobileNumber (n)
age (n)
DOB (date)

candidate key columns

In db table we can have 0 or more candidate key columns.

PK, FK, NNK ,UK and etc... physical/pratical keys of Db table i.e they can be added in Db tables..

=>physical keys can be added pratically in db table creation
=>Logical keys can not be added pratically in db table creation.. They help to choose proper physical key..

Natural key Column (Logical Keys)

=> The candiate key column whose values are having business meaning , expected from endusers and will change becoz of outside business policies changes is called Natural key column

eg:: aadharNo, panNo , voterId , drivingLicense , mobileNo , bankAccountNo and etc...

=>These columns values are natually ID values inside and outSide Db s/w and can be used to identify and access records.

Surrogate Key Column (Logical keys)

=>The candiate key column whose values do not have any business meaning and generated by underlying Db s/w or App dynamically is called surrogate key column.
eg :: sno generated by App of School , eno generated by Employeer and etc...
-> in oracle Db s/w we can use sequence support to generate colum values dynamically
-> in mysql Db s/w we can Indentity column (column with AutoIncrement constraint) support for the same
-> Id Generators concept hibernate..
->Different algorithms used in App..
as

Limitations of taking PK colum Natural key column

=> values are going to be very lengthy ,So more memory is required
=> Values are having buniess meaning..i.e they will change based on real world policies an effects other dependent db tables and relavent java code (The change is very costly)
=> Excepted from end users .. If endusers fails give the records insertions will be failed.

Advantages of taking PK column as Surrogate keycolumns

=> Values are shorter values .. So they use less memory
=> Values do not have any business meaning .. they will not be chanaged becoz real world policy chanages
=> Not expected from endusers.. will be generated by the underlying App or Db s/w dynamically..

Proceudre to use sequence in oracle to generated PK colum (sno) value dynamically

step1) make sure the u r insert opeation is ready having PreparedStatement obj..

step2) create sequence in oracle

```
SQL> create sequence sno_seq1 start with 1000 increment by 1;
```

step3) Link the above sequence in INSERT SQL query creation

```
private static final String STUDENT_INSERT_QUERY="INSERT INTO STUDENT VALUES(SNO_SEQ1.NEXTVAL,?,?,?)";
```

step4) Develope the Rest of the App in regular manner..

PsInsertTestSurrogatePK.java

```
package com.nt.jdbc1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

public class PsInsertTestSurrogatePK {
  private static final String STUDENT_INSERT_QUERY="INSERT INTO STUDENT VALUES(SNO_SEQ1.NEXTVAL,?,?,?)";

    public static void main(String[] args) {
        Scanner sc=null;
        Connection con=null;
        PreparedStatement ps=null;
         try {
              //read inputs
              sc=new Scanner(System.in);
              int count=0;
              if(sc!=null) {
                   System.out.println("enter students count::");
                   count=sc.nextInt();
              }
              //register JDBC driver
               //Class.forName("oracle.jdbc.driver.OracleDriver");

              //establish the connection
              con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");

              //create PreparedStatement object having pre-compiled SQL query
              if(con!=null)
                ps=con.prepareStatement(STUDENT_INSERT_QUERY);

              //read input values from enduser ,set them to query param values and execute the pre-compiled
              //SQL query for multiple times
              if(ps!=null && sc!=null) {
                   for(int i=1;i<=count;++i) {
                        //read each student input values
                        System.out.println("enter "+i+" student details");
                        System.out.println("enter student name::");
                        String name=sc.next();
                        System.out.println("enter student address::");
                        String addrs=sc.next();
                        System.out.println("enter student avg::");
                        float avg=sc.nextFloat();
                        //set each student details as pre-compiled SQL query params
                         ps.setString(2, name); ps.setString(2, addrs); ps.setFloat(3, avg);
                        //execute pre-compiled SQL query each time
                        int result=ps.executeUpdate();
                        //process execution result of pre-compiled-SQL query
                        if(result==0)
                             System.out.println(i+" student details not inserted");
                        else
                             System.out.println(i+" student details are inserted");
                   }//for
              }//if

         }//try
         catch(SQLException se) {
              se.printStackTrace();
         }
         catch(Exception e) {
              e.printStackTrace();
         }
         finally {
              //close jdbc objs
              try {
                   if(ps!=null)
                        ps.close();
              }
              catch(SQLException se) {
                   se.printStackTrace();
              }
              try {
                   if(con!=null)
                        con.close();
              }
              catch(SQLException se) {
                   se.printStackTrace();
              }
              try {
                   if(sc!=null)
                        sc.close();
              }
              catch(Exception e) {
                   e.printStackTrace();
              }
         }//finally

    }//main
}//class
```

Procedure to make Surrogate key column as PK col in MYSQL Db s/w (Using autoincrement constraint based Indentity column)

note:: mysql does not support sequences.. so apply auto increment constraint on PK column

step1) make the "sno" column of student db table as Identity column by applying auto_increment constraint..

```
mysql> alter table student modify sno int auto_increment ;
```

step2) change SQL query in the Java App as shown below

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

=>oracle sequence and mysql identity col genates the next id value using <previous value> +1 formulae..

step3) Develop the remaining app in regualar using Connector/J JDBc driver

=>same as above App just chage SQL Query and JDBC driver details..

=>oracle does not support identity columns upto 11g
=>mysql does not support sequences..

=> Auto increment constraint can be applied only on PK column i.e one db table can have only one autoincrement constraint column.

What is difference b/w static SQL query and Dynamic SQL query?

Ans) the SQL Query with out input values or the SQL query with hardcoded input values
either directly or through variables is called static SQL query

eg1:: SELECT SNO,SNAME,SADD FROM STUDENT
eg2:: SELECT SNO,SNAME,SADD FROM STUDENT WHERE SNO>=100 AND SNO<=200
eg3:: String query="SELECT SNO,SNAME,SADD FROM STUDENT WHERE SNO>="+start+" AND SNO<="+end;
=>start,and are variables.

The SQL query that takes parameters(?) representing inputs and allows to set values to parameters(?) after compilation is called Dynamic SQL query.
or Query

eg1: SELECT SNO,SNAME,SADD FROM STUDENT WHERE SNO>=? AND SNO<=?
eg2: UPDATE STUDENT SET SNAME=? AND SADD=? WHERE SNO=?
eg3: INSERT INTO STUDENT VALUES(?,?,?,?)

=> Simple statement object deals with only static SQL queries
=> PreparedStatement object can deal with both static and Dynamic SQL queries
=> Pre-compiled SQL query can be a static SQL Query or Dynamic SQL query with or with out parameters or input values

eg1: PreparedStaement ps1=con.prepareStaatement("SELECT * FROM STUDENT"); /makes the static SQL query as pre-compiled SQL query.

eg2: PreparedStaement ps2=con.prepareStaatement("SELECT * FROM STUDENT WHERE SNO>=100 AND SNO<=200");
//makes the static SQL query with hard coded inputs as pre-compiled SQL query.

eg3: PreparedStaement ps2=con.prepareStaatement("SELECT * FROM STUDENT WHERE SNO>=? AND SNO<=?");
//makes the dynamic SQL query as pre-compiled SQL query.

=>pre-compiled SQL can be taken with or with out params i.e pre-compiled SQL Query can be a static SQL query or dynamic SQL query.

=>Parameters in Dynamic SQL query represented PreparedStatement obj can represent only input values i.e they can not represent db table names, col names, SQL key words and etc..

eg1:: SELECT SNO,SNAME,SADD FROM STUDENT WHERE SNO>=? ANO SNO<=? (valid)
eg2:: SELECT SNO,SNAME,SADD FROM STUDENT WHERE ?>=? AND ?<=? (invalid) col names can not be taken as params
eg:: SELECT SNO,SNAME,SADD FROM STUDENT ? SNO>=? AND SNO<=? (invalid) SQL keyword can not be taken as params
eg4:: SELECT SNO,SNAME,SADD FROM_1 WHERE SNO>=? AND SNO<=? (invalid) db table name can not be as param
eg5:: SELECT ?,?,? FROM STUDENT WHERE SNO>=? AND SNO<=? (invalid) col names can not be taken as params

---

=> Java compiler compiles java code by verifying the syntax with the support of java keywords that are present in java code
=> DB engg compiles SQL queries by verifying the syntax with the support of SQL keywords that are present in SQL query.

If every thing of SQL query is allowed as parmeter(?) than people may write the SQL query as shown below
?,?,?,?,?,?,? then the compilation of this SQL query becomes impossible.

=>In realtime the developers generally do not use SQL prompts to send and execute SQL query in Db s/w rather they prefer working with GUI DB tools for the same .. becoz this process does not need strong SQL knowlege.

examples SQL developer for oracle ,TOAD for Oracle , TOAD for mysql , Mysql Workbench , SQLYOG for mysql, PGAdmin for PostgreSQL , PL/SQL Developer and etc.

note:: Certails GUI DB tools will comes along with Dbs/w installation (eg: mysql work bench , pgAdmin and etc..). where as few others must be installed explicitly (eg: SQL developer for oracle, TOAD for oracle and etc..)

Working with SQL DEveloper to create DB table having multiple columns

step1) install SQL developer
(download as zip and extract it for installation)

step2) Launch SQL Developer and create a conneciton with Oracle DB s/w

=>use sqldeveloper.exe file from the above zip file extraction (create short cut on desktop)
=> For connection creation ---> use + or newe connection and fill the details

step3) create new DB table

expand con1 ----> right click on tables ----> new --->

--->ok -.

Generated SQL query

```
CREATE TABLE "SYSTEM"."PERSON_INFO_DATES"
   (    "PID" NUMBER(10,0) NOT NULL ENABLE,
        "PNAME" VARCHAR2(20 BYTE),
        "DOB" DATE,
        "DOJ" DATE,
        "DOM" DATE,
        CONSTRAINT "PERSON_INFO_DATES_PK" PRIMARY KEY ("PID")) ;
```

Two types of DB s/ws

(1) SQL DB s/ws =>> stores data as db table rows and cols
=> useful if the data is formatted and fixed size info in each record
eg: oracle ,mysql, postgreSQL and etc.. (70%)

(1) No-SQL DB s/ws =>> stores data as documents
=> useful if the data is not formatted and dynamically growable info each record
eg: MongoDB ,Cassendra, Neo4j and etc.. (30%)

Procedure to create Logical DB having DB table in mysql Db s/w using MYSQL Work bench

note: MYSQL Workbench comes along with mysql Db s/w installation..

step1) make sure that mysql Db s/w installed.

n

step2) Lauch mysql workbench

=>search for mysql workbench and launch it

step3) Establish the connection

⊕ - add con

TEst Connection ---->ok

step4) Create Logical DB in mysql DB s/w..

name :: NTAJ415DB1 ---> apply----> ok

step5) create DB table in logical DB NTAJ415DB1

Expand ---> NTAJ415DB1--- right click tables ----> new --->

==> apply -->next -->

```
CREATE TABLE `ntaj415db1`.`person_info_dates` (
  `pid` INT NOT NULL AUTO_INCREMENT,
  `pname` VARCHAR(15) NULL,
  `DOB` DATE NULL,
  `DOJ` DATE NULL,
  `DOM` DATE NULL,
  PRIMARY KEY (`pid`),
  UNIQUE INDEX `pid_UNIQUE` (`pid` ASC) VISIBLE);   ====>Finish.
```

Working with Date values

=>while dealing DOB,DOJ, DOM , billDate , payment Date , calculating age and etc.. we need insert and retrive date values..

=>Do not store date values in Db table as String column values becoz we can not perform arithmetic and logical opeartions on date values

=>Always prefer stroing date values in the db table columns whose data type is "date"..

=> All most ,all major Db s/ws support "date" data type... but internally maintain date values in different patterns.

oracle date pattern :: dd-MMM-yy eg: 20- OCT-90

mysql date pattern :: yyyy-MM-dd eg: 1990-10-20

to

=> we can insert date values to DB table columns whose date types is "date" by using Simple statement obj but we need pass date value directly in SQL query that to in the pattern that is supported by the underlying Db s/w which very complex becoz endusers belongs to different countries prefer giving date values in different patterns .. and Db s/w expects on other patterns..

indian date pattern :: dd-MM-yyyy eg:: 20-10-1990
US Citizen date pattern :: MM-dd-yyyy eg:: 10-20-1990
china date pattern :: yyyy-MM-DD eg:: 1990-10-20

=>To over come problems use PreparedStatement by taking parameters (?) for date values in pre-compiled SQL query and by setting date values to parameters using ps.setDate(-,-) method in form java.sql.Date class objs.

=>Once we set date value to SQL query parameter as java.sql.Date class obj using ps.setDate(-,-) ..the JDBC driver takes the resposibility of inserting date value to DB table col in the pattern that is supported by underlying Db s/w.

what is difference b/w java.util.Date class and java.sql.Date class?

java.util.Date(c)
↑ extends
java.sql.Date(c)

java.util.Date

(a) Not suitable for DB CURD operations becoz contains multiple details as english para statements other

(b) Maintains date , time and multiple details

(c) if we create this object using 0-param constructor we get system date and time

(d) can not be used as query param value while working ps.setDate(-,-) method

java.sql.Date

(a) Suitable for Db CURD opeartions

(b) just maintain only date value .. does not maitain any time info other info

(c) There is no such provision. becos there is no 0-param constructor

(d) can be used as query param value while working with ps.setDate(-,-) method

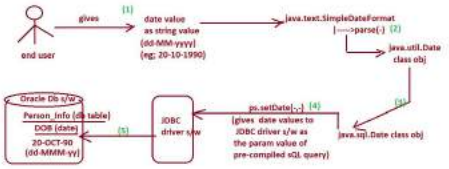
note1:: most of the methods in java.util.Date class are deprected .. So as alternate use java.util.Calendar or java8 date and time api

note:: From Java 8 onwords 3 more classes are given to work date and time as part java8 date and time api

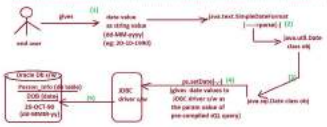
LocalDate :: To work date values
LocalTime :: to work time values
LocalDateTime :: to work both date,time values.

=>As of now JDBC4.x+ is not supporting java8 date and time api toward insertion or retrieving opeations of date and time api.

DB s/w and JDBC drlver s/w idenpendent procedure to insert Date values to Date data types col of Db table



DB s/w and JDBC driver s/w liberpendent procedure to insert Date values to Date data types col of Db tables




```
//converting String date value to java.util.Date class obj      parse( ) method takes  it  pattern of
    String str="31-11-1990" ; //dd-MM-yyyy                                  input string date value
    SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
    java.util.Date sd1=sdf.parse(s1);
    System.out.println("String date value :"+s1);
    System.out.println("util date :"+sd1);
```

note: The parse( ) of java.text.SimpleDateFormat is given to convert given specified format string date value
to java.util.Date class object and the format( ) of java.text.SimpleDateFormat class performs the reverse.
(java.util.Date class obj to String date value)

```
//converting  java.util.Date class obj to java.sql.Date class obj
        long ms=sd1.getTime();  // gives no.of milliseconds  that  elapsed b/w
                                (util date and time and  1970 jan 1st mid night 00.00 hrs  (Epoch) standard)
        java.sql.Date sd2=new java.sql.Date(ms);
        System.out.println("util date :"+sd1);
        System.out.println("sql date :"+sd2);
```

==in any programing language or OS or DB s/w, the date and time values are internally maintained as milliseconds with respect to
jan 1st 1970 midnight 00.00 hours.

note:- while working with the parse( ) method of java.text.SimpleDateFormat class if pass extra days in date value, extra months in
month value then will be adjusted as during conversion....

```
            41-22-1990  [dd-MM-yyyy] ---nomes as 10  nov 1991
```

```
// If String date value pattern  is yyyy-MM-dd pattern then it can be converted directly to java.sql.Date class obj
    // with out converting to  java.util.Date class obj
    String s2="1991-12-15" ;  //yyyy-MM-dd
    java.sql.Date sd3=java.sql.Date.valueOf(s2);          valueOf( ) is static method of  java.sql.Date class
    System.out.println("String date value :"+s2);
    System.out.println("sql date value :"+sd3);          ==>default date pattern of java.sql.Date class obj is " yyyy-MM-dd"
```

==if extra days in Date value and extra months in moth values will not be adjusted as part
conversion ... rather we will get Exception in thread "main" java.lang.IllegalArgumentException

To create sequence in oracle using SQL developer

expand conn ---> right click on sequences---> new sequence

generated SQL :-
CREATE SEQUENCE "SYSTEM"."PID_SEQ" MINVALUE 100 MAXVALUE 10000 INCREMENT BY 1 START WITH 300 CACHE 20 NOORDER
NOCYCLE ;

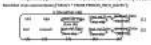
PsDateInsertTestOracle .java

```
package com.nt.jdbc;

import java.sql.Connection;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Scanner;

/*CREATE TABLE "SYSTEM"."PERSON_INFO_DATES"
(    "PID" NUMBER(3,0) NOT NULL ENABLE,
 "PNAME" VARCHAR2(30 BYTE),
 "DOB" DATE,
 "DOJ" DATE,
 "DOM" DATE,
 CONSTRAINT "PERSON_INFO_DATES_PK" PRIMARY KEY ("PID") )*/

public class PsDateInsertTestOracle {
 private static final String INSERT_DATE_QUERY="INSERT INTO PERSON_INFO_DATES VALUES(PID_SEQ.NEXTVAL,?,?,?,?)";
    public static void main(String[] args) {
        Scanner scan=null;
        Connection con=null;
        PreparedStatement ps=null;
            try {
                //read inputs
                scan=new Scanner(System.in);
                String name=null, sdobval=null,sdojval=null,sdomval=null;
                if(scan!=null) {
                    System.out.println("Person name:");
                    name=scan.next();
                    System.out.println("Person DOB(dd-MM-yyyy) :");
                    sdobval=scan.next();
                    System.out.println("Person DOJ(yyyy-MM-dd) :");
                    sdojval=scan.next();
                    System.out.println("Person DOM(MMM-dd-yyyy) :");
                    sdomval=scan.next();
                }
                //convert  String date values to java.sql.Date class objs
                 //For DOB(dd-MM-yyyy)
                    //convert String date value to java.util.Date class obj
                    SimpleDateFormat sdf1=new SimpleDateFormat("dd-MM-yyyy");
                    java.util.Date sdobval1=sdf1.parse(sdobval);
                    //converting java.util.Date class obj to  java.sql.Date class obj
                    long ms=sdobval1.getTime();
                    java.sql.Date sqldob=new java.sql.Date(ms);
                //For DOJ  (yyyy-MM-dd)  -direct conversion )
                            //converting String date value  to  java.sql.Date class obj
                            java.sql.Date sqldoj=java.sql.Date.valueOf(sdojval);
            //For DOM(MMM-dd-yyyy)
                                //convert String date value to java.util.Date class obj
                                SimpleDateFormat sdf2=new SimpleDateFormat("MMM-dd-yyyy");
                                java.util.Date sdomval2=sdf2.parse(sdomval);
                                //converting java.util.Date class obj to  java.sql.Date class obj
                                 ms=sdomval2.getTime();
                                java.sql.Date sqldom=new java.sql.Date(ms);

                            //Load JDBC driver class
                                Class.forName("oracle.jdbc.driver.OracleDriver");
            //establish the connection
                                con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                            //create  PreparedStatement obj
                                if(con!=null)
                                    ps=con.prepareStatement(INSERT_DATE_QUERY);
                                //set values to query parameters
                                if(ps!=null) {
                                    ps.setString(1, name);
                                    ps.setDate(2,sqldob);
                                    ps.setDate(3,sqldoj);
                                    ps.setDate(4, sqldom);
                                }
                                //execute  query
                                int count=0;
                                if(ps!=null)
                                    count=ps.executeUpdate();
                                //process the results
                                if(count==0)
                                    System.out.println("Record not inserted");
                                else
                                    System.out.println("Record inserted");
            }//try
            catch(SQLException se) {
                se.printStackTrace();
                System.out.println("Problem in record insertion");
            }
            catch(Exception e) {
                e.printStackTrace();
            }
            finally {
                //close jdbc objs
                try {
                        if(ps!=null)
                            ps.close();
                    }
                    catch(SQLException se) {
                        se.printStackTrace();
                    }

                    try {
                            if(con!=null)
                                con.close();
                    }
                    catch(SQLException se) {
                        se.printStackTrace();
                    }

                    try {
                            if(scan!=null)
                                scan.close();
                    }
                    catch(Exception e) {
                        e.printStackTrace();
                    }

            }//finally

    }//main
}//class
```

Illustrating the data nature from DB tuples out.

- main run one writes single impression data in thread-determined obj by continuing data values.
- main run two reads impression data from thread-determined obj by continuing data values. If impression dictionary obj has a processed ID in the pro.readyMultiUser query and also an out .pro.out there, also move the data query given value.
- When data exceeding, read data impression as type owner. In the loop, thread-saving data grabbing using Single Developer data to associate the DB values.
- In office executing, note doubt impress given data values. , call grabbed.thisWriter old binding .save.up=data data data representing the removed data values. , And the run thisun data values differs using in getOrDefault() methods.
or , in getOrDefault() methods.

```
==>.getString() .gives.data values . or String . to the pointer . .grip.data.out
==>.getInteger() .gives.data values . or .java.io.Data.i.see.obj . fueling the pointer . .grip.defined
ResultSet .in.pro.connectionData ("Select * FROM PRIMARY_file to_pull IN");
```


1 (ResultSet) obj
NO
NAME
Data.obj.for_getString().gives.data.values
Data.obj.for_getInteger().gives.data.values
Data.obj.for_getDouble().gives.data.values
Data.obj.for_getBoolean().gives.data.values
Data.obj.for_getObject().gives.data.values
Data.obj.for_getBigDecimal().gives.data.values
Data.obj.for_getBytes().gives.data.values
Data.obj.for_getDate().gives.data.values
Data.obj.for_getTime().gives.data.values
Data.obj.for_getTimestamp().gives.data.values
Data.obj.for_getAsciiStream().gives.data.values
Data.obj.for_getUnicodeStream().gives.data.values
Data.obj.for_getBinaryStream().gives.data.values
Data.obj.for_getCharacterStream().gives.data.values
Data.obj.for_getRef().gives.data.values
Data.obj.for_getBlob().gives.data.values
Data.obj.for_getClob().gives.data.values
Data.obj.for_getArray().gives.data.values
Data.obj.for_getURL().gives.data.values
Data.obj.for_getRowId().gives.data.values
Data.obj.for_getNClob().gives.data.values
Data.obj.for_getSQLXML().gives.data.values
Data.obj.for_getNString().gives.data.values
Data.obj.for_getNCharacterStream().gives.data.values
Data.obj.for_getObject().gives.data.values
Data.obj.for_getObject().gives.data.values
Data.obj.for_getObject

```
app.controller('ctrl_name', function($s
  // Controller for ctrl_name
  $scope.ctrl_name = {};
  var vm = $scope.ctrl_name;
  vm.title = 'ctrl_name';
  vm.items = [];
  vm.selected = null;
  vm.loading = false;
  vm.error = '';
  vm.load = function() {
    vm.loading = true;
    vm.error = '';
    $http.get('/api/ctrl_name')
      .then(function(response) {
        vm.items = response.data;
        vm.loading = false;
      })
      .catch(function(err) {
        vm.error = err.message;
        vm.loading = false;
      });
  };
  vm.select = function(item) {
    vm.selected = item;
  };
  vm.load();
});
```

```
app.directive('d3_scatter', ['$parse', function($parse) {
  return {
    restrict: 'E',
    replace: true,
    template: '<svg></svg>',
    link: function(scope, element, attrs) {
      var data = $parse(attrs.data)(scope);
      var width = attrs.width || 600;
      var height = attrs.height || 400;
      var svg = d3.select(element[0])
        .attr('width', width)
        .attr('height', height);
      var x = d3.scale.linear()
        .domain(d3.extent(data, function(d) { return d.x; }))
        .range([0, width]);
      var y = d3.scale.linear()
        .domain(d3.extent(data, function(d) { return d.y; }))
        .range([height, 0]);
      svg.selectAll('circle')
        .data(data)
        .enter()
        .append('circle')
        .attr('cx', function(d) { return x(d.x); })
        .attr('cy', function(d) { return y(d.y); })
        .attr('r', 4);
    }
  };
}]);
```

Interacting multiple DB s/w from JDBC App

[illegible]

[illegible]

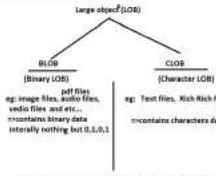
try with resource

[illegible]

public interface AutoCloseable

[illegible]

Working with Large objects (files)
==========



pdf files
eg: image files, audio files, vedio files and etc...
=>contains binary data interally nothing but 0,1,0,1

eg: Text files, Rich Rich Files , word docs,csv files and etc...
=>contains characters data nothing but alphbets,numbers, symbols and etc..

=>While working SimpleStatement obj we need to place input values directly in the SQL query but we can not place BLOB, CLOB files content directly in the SQL query., so we can not use Simple Statement obj insert BLOB values to Db table cols

=>PreparedStatement's pre-compiled SQL query contains parameters[?] and we set streams representing BLOB , CLOB files as query param param values.. This makes inserting BLOB, CLOB values to DB table cols quiet easy using PreperedStatement obj.

=> While developing matrimony Apps, Job portal Apps, Social network Apps, Profile mgmt and etc. Apps we need to perform insertion and retriving opreations of LOBs..
=> All major DB s/ws like oracle,mysql and etc.. are giving support Large objects by giving appropriate data types like BLOB,CLOB and etc..

=>A stream is continous of flow data b/w source and Destination
=> Source can use stream to write data to destination and/or to read data from destination

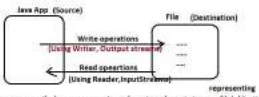
There are two types of stream s in java based on the kind of data they read/write
a) Binary Streams /Byte Streams
=> Useful to read/write binary data and text data from/to destination
eg:: FileInputStream, FileOutoutStream, DataInputStream , DataOutputStream and etc..
=> Inputs streams are there to read data from Destination
=> output streams are there to write data to destination

A stream can have different destiations like file , collection, object, array, key board, console,network(socket) and etc..

=>System.in is a built-in input stream pointing to keyboard destination
=>System.out is a built-in output stream pointing to console/monitor destination
=>System.err is a built-in error stream pointing to console/monitor destination

b) Character streams
=>Useful to read/write only character data from/to destination
eg:: FileReader , FileWriter , BufferedReader , BufferedWriter and etc...

Reader streams are there to read data from destinations.
Writer streams are there to write data to destinations.



We set Streams as pre-compiled query parameter values in order set stream file's/destination's data as query param value
For that we can use
ps.setBinaryStream(-,-) or ps.setBlob(-,-)
ps.setCharacterStream(-,-) or ps.setClob(-,-)

JDBC driver and DB s/w indepent procedure to insert BLOB Value (image file) to DB table BLOB data type col value
==========

step1) create PreparedStatement object having pre-compiled SQL query

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

step2) create InputStream pointing to BLOB file (image file)

```
InputStream isrnew FileInputStream("e:\\photos\\rashmika.jpg");
```

as sequence character

\" ==> gives printable "
\' ==>gives printable '
\\ ==>gives printable \

S.o.p(" \" "); prints "
S.o.p(" \\ "); prints \

step3) set pre-compiled query param values

```
ps.setString(1, "rashmika mandana");
ps.setString(2, "blore");
ps.setBinaryStream(3,is); (or)
ps.setBlob(3,is);
```

step4) execute the Query

```
int count=ps.executeUpdate();
```

step5) close strems

```
ps.close();
con.close();
is.close();
```

rashmika.jpg (e:\photos)

Example App
==========

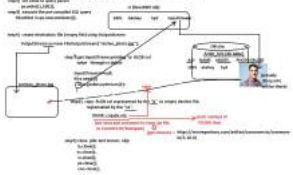
```
package com.nit.jdbc1;

import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

/*CREATE TABLE "SYSTEM"."ARTIST_INFO"
(    "AID" NUMBER(10,0) NOT NULL ENABLE,
     "NAME" VARCHAR2(20 BYTE),
     "ADDRS" VARCHAR2(20 BYTE),
     "PHOTO" BLOB,
     CONSTRAINT "ARTIST_INFO_PK" PRIMARY KEY ("AID"))

  CREATE SEQUENCE  "SYSTEM"."AID_SEQ"  MINVALUE 1000 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 1000 CACHE 20 NOORDER  NOCYCLE ;

    */

public class PsBlobInsertTest {
  private static final String  INSERT_ARTIST_QUERY="INSERT INTO ARTIST_INFO VALUES(AID_SEQ.NEXTVAL,?,?,?)";
     public static void main(String[] args) {
          try(Scanner sc=new Scanner(System.in)){
               //read inputs
               String name=null,addrs=null, photoLocation=null;
               if(sc!=null) {
                    System.out.println("Enter artist name::");
                    name=sc.next();
                    System.out.println("Enter Artist address::");
                    addrs=sc.next();
                    System.out.println("Enter Artist Location::");
                    photoLocation=sc.next();
               }//if
               //create InputStream pointing photo file
               try(InputStream is=new FileInputStream(photoLocation)){
                    //establish the connection and prepared Statement object
                    try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
                      PreparedStatement ps=con.prepareStatement(INSERT_ARTIST_QUERY);      )     {
                    //set values to query param
                    if(ps!=null) {
                         ps.setString(1,name);
                         ps.setString(2, addrs);
                         ps.setBinaryStream(3, is);
                    }
                    //execute the query
                    int count=0;
                    if(ps!=null)
                         count=ps.executeUpdate();

                    //process the results
                    if(count==0)
                         System.out.println("record not inserted");
                    else
                         System.out.println("Record inserted");
               }//try3
               }//try2
          }//try1
        catch(SQLException se) {
             se.printStackTrace();
             System.out.println("Problem in record insertion");
        }
     catch(Exception e) {
          e.printStackTrace();
     }

     }//main
}//class
```

```
>>> ALIAS(5) DB size ... we have  BLOB, CLOB thick Apply difficulting new Agt meaning No each areas of scheme
>> in mysqlTB a be we trusting multiple welfare of BLOB, CLOB data types
                    TINYBLOB, BLOB, MEDIUM BLOB, LONG BLOB  and mo...
                    And MyS. that  .setAsciiStream  .calculated

    package com.ch.pl8.v1;

    import java.io.BufferedReader;
    import java.io.InputStream;
    import java.sql.Connection;
    import java.sql.DriverManager;
    import java.sql.PreparedStatement;
    import java.sql.ResultSet;
    import java.util.Scanner;

    /*CREATE TABLE IMG(ID INT, name_info );
                    age  INT NOT NULL ID_NO ,GENDER BIT,
                    mother  VARCHAR(20) NO NULL,
                    photo1  VARCHAR(1024)NULL,
                    photo2  LONGBLOB NULL,
                    RESIDENT BIT ( 1bit ),
                    checkout  Blank   sex_checkout  ( null  one2 whatever)  */

    public class PutMaterialsIMG{
        private static final String  SELECT_DATA = "insert into ARTIST_INFO(NAME,NAMEUPHOTO,VOLUME)VALUES(?,?,?)";
            public static void main(String[] args){
                try(Scanner sc = new Scanner(System.in)){
                    System.out.println("Enter name");
                    String name=sc.nextLine();
                    if(name.equals("")){
                        System.out.println("Enter artist name.");
                    }else{
                        System.out.println("Enter Artist address.");
                        address=sc.next();
                        System.out.println("Enter Artist photo location.");
                        photo=sc.next();
                        System.out.println("Enter Artist volume");
                        int volume=sc.nextInt();
                        //Connection inputStream=null;
                        try(InputStream is=new FileInputStream(photo);){
                            //establishes connection and prepared statement object
                            try(Connection con=DriverManager.getConnection("jdbc:mysql://localhost/" + "test", "root",
                                PreparedStatement ps=con.prepareStatement(INSERT_ARTIST_BLOB);    ){
                                //setting the query params
                                try(ResultSet
                                    ps.setString(1, name);
                                    ps.setString(2, address);
                                    ps.setBinaryStream(3, is);
                                ){
                                //execute the query
                                int count=
                                try{result=
                                    count=ps.executeUpdate();
                                }
                                //process the results
                                if(count==0)
                                    System.out.println("record not inserted");
                                else
                                    System.out.println("record inserted");
                            }catch
                            }catch
                        }catch
                    }catch(FileNotFoundException e1 ){
                        System.out.println("Incorrect or no such location");
                    }
                }catch(Exception e2 ){
                    e.printStackTrace();
                }

            }//main
    }//class
```

Retrieving BLOB values from DB table use

- Retrieving BLOB values from DB table use
- main use case where images stored in object (query=sc.nextLine(); or PreparedStatement obj1(query with conditions) to retrieve
  data values from DB table use.
- In Java BLOB data to values you retrieved use(stored into file object Store use use as ps.getBinaryStream(1) (input InputStream retrieving.
  from DB data.

```
    DB  SQL      stmt  JDBC  Driver      Independent     Procedures  to retrieve BLOB values from DB table use.
    ------------------------------------------------------------------------------------------------------------

    class C1 create PreparedStatement object having pre-compiled SQL (SELECT) Query
        PreparedStatement ps=
            con.prepareStatement("SELECT VOLUME,NAMEUPHOTO FROM ARTIST_INFO WHERE ID=?");
    step2  set value to the query param
        ps.setInt(1,101);
    step3  execute the pre-compiled SQL query
        ResultSet rs=ps.executeQuery();
```




```
    step4  navigate  the  result  using  loop(while/do)
        while(rs.next()){
            //get the values from the result set
            int id=rs.getInt(1);
            String name=rs.getString(2);
            Blob b=rs.getBlob(3);
            //read data from BLOB object
            InputStream is=b.getBinaryStream();
            //create FileOutputStream object
            FileOutputStream fos=new FileOutputStream("artists_photo.jpg");
            //read data from is and write to fos
            byte[] buffer=new byte[1024];
            while(is.read(buffer)>0){
                fos.write(buffer);
            }
            System.out.println("Record is inserted");
        }
    step5  close  JDBC  object  variables
        rs.close();
        ps.close();
        con.close();
        stmt.close();
        //is.close();
        //fos.close();
    }catch(Exception e){
        e.printStackTrace();
    }
}
```


```
/*
public class RetrieveMaterialsIntoMySQL {
    private static final String  ARTIST_RETRIEVAL_QUERY="SELECT VOLUME,NAMEUPHOTO FROM ARTIST_INFO WHERE ID=?";
        public static void main(String[] args){
            try(Scanner sc=new Scanner(System.in)){
                System.out.println("Enter Artist No.");
                int no=sc.nextInt();
                try(Connection con=DriverManager.getConnection("jdbc:mysql://localhost/" + "test", "root",
                                PreparedStatement ps=con.prepareStatement(ARTIST_RETRIEVAL_QUERY);
                                ){
                            //set query params
                            ps.setInt(1, no);
                            //execute the query
                            try(ResultSet rs=ps.executeQuery()){
                                //get the values from the result set
                                if(rs.next()){
                                    //read BLOB data
                                    Blob b=rs.getBlob(3);
                                    String name=rs.getString(2);
                                    //get InputStream object pointing to BLOB
                                    InputStream is=b.getBinaryStream();
                                    //create FileOutputStream object pointing to artistsphoto.jpg
                                    FileOutputStream fos=new FileOutputStream("retrieve_image.jpg");
                                    byte[] buffer=new byte[1024];
                                    while(is.read(buffer)>0){
                                        fos.write(buffer);
                                    }
                                    // copy BLOB properties to ClientInfo file
                                    clientInfo.append(name);
                                    System.out.println("file info entry is continued and stored in the file");
                                }//if.inf
                                else{
                                    //else if
                                    System.out.println("Record not found");
                                }//else
                                //if.inf
                            }//try.rs closing
                        }//try.ps  close
                    }//try.con
                    catch(SQLException e){
                        e.printStackTrace();
                    }
                    catch(Exception e1 ){
                        e.printStackTrace();
                    }
                }//main
            }//class
```

Assignment

- Write a JDBC App to insert Img/Vid/Audio in standalone/WebEntity App
  - Ex: student gender SSN photo resume DOJ,Vid/DB thumbnail(s) audio/clip website/s

  - Ex: Artist gender SSN photo resume DOJ,Vid/DB thumbnail(s) audio/clip website/s

- Write a JDBC app to retrieving same details from Mini/Enterp App

```
Font Tabs, 2014 Excel has various, any, called 35,000 but not so a password. these files allow using
Handle books in Reader Amazing.

>> To use 1,000 this shared. Reader to your computer(100) assignments notice not need to use
>> and Password to work. Use as well Files, Just Word. The Over 10,000 value.

>> To get Outside Screen subject: word missing. 2,000 of yellow to get this photo object. Use as can
>> get Chrome but Cannot Creative. Others Removing 2,134 Word

>>  In word Who Use, can do not have Excel 10,000 this Apps., others we have. TRATAR ANDROID and ANDROID
    these Apps as alternative to 2,000 Java App.

>> 50 Size to lower of Reading., word words no others for Internet. from icons found Internet 50
>> 100 Size Access of writing., to Access no others for Internet. from icons to Image Processing
>> 127 character. Instead of Writing., to Access to Authority. Internet from java, to Android 5.1
>> 127 Character. Instead of. Writing., to Access to Authority. Internet from Java, to Android 5.1


  /*CREATE TABLE "SYSTEM"."EMPLOYEE_DEMO"
   (    "NO" NUMBER(22,0) NOT NULL ENABLE,
        "NAME" VARCHAR2(30 BYTE),
        "ADDRR" VARCHAR2(40 BYTE),
        "SALARY" NUMBER(22,0),
        CONSTRAINT "EMPLOYEE_DEMO_PK" PRIMARY KEY ("NO")

  CREATE INDEX "SYSTEM"."EMP_IDX" ON EMPLOYEE_DEMO (NAME, ADDRR) COMPUTE STATISTICS;
  */

public class MCDBConnectionTest {
    private static final String INSERT_EMPLOYEE_QUERY="INSERT INTO EMPLOYEE_DEMO VALUES(?,?,?,?)";
    public static void main(String[] args) {
        try{Class.forName("com.mysql.jdbc.Driver");
            try(Scanner in=new Scanner(System.in)){  //try1
                //Read Inputs
                String name=null,addr=null,salary=null,no=null;
                try{
                    System.out.println("Enter Employee name: ");
                    name=in.next();
                    System.out.println("Enter Employee address: ");
                    addr=in.next();
                    System.out.println("Enter Employee salary: ");
                    salary=in.next();
                    System.out.println("Enter Employee number: ");
                    no=in.next();
                }catch(InputMismatchException ie){
                    System.out.println("Invalid Input");
                }
                //Create Connection and Statement
                try(Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/","root","root");
                    PreparedStatement ps=con.prepareStatement(INSERT_EMPLOYEE_QUERY);){  //try2
                    //Set values to query
                    ps.setString(1,name);
                    ps.setString(2, addr);
                    ps.setDouble(3,Double.parseDouble(salary));
                    ps.setInt(4,Integer.parseInt(no));
                    //Execute the query
                    int count=ps.executeUpdate();
                    //process the results
                    if(count==0)
                        System.out.println("record not inserted");
                    else
                        System.out.println("Record Inserted");
                }//try2
            }//try1
        }catch(ClassNotFoundException cnf){
            cnf.printStackTrace();
            System.out.println("Problem in record insertion");
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}//class
```

```
public class MVCJdbcInsertTest {
    private static final String INSERT_EMPLOYEE_QUERY="INSERT INTO EMPLOYEE_DEMO VALUES(?,?,?,?)";
    public static void main(String[] args) {
        try{Class.forName("com.mysql.jdbc.Driver");
            try(Scanner in=new Scanner(System.in)){  //try1
                //Read Inputs
                String name=null,addr=null,salary=null,no=null;
                try{
                    System.out.println("Enter Employee name: ");
                    name=in.next();
                    System.out.println("Enter Employee address: ");
                    addr=in.next();
                    System.out.println("Enter Employee salary: ");
                    salary=in.next();
                    System.out.println("Enter Employee number: ");
                    no=in.next();
                }catch(InputMismatchException ie){
                    System.out.println("Invalid Input");
                }
                //Create Connection and Statement
                try(Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/","root","root");
                    PreparedStatement ps=con.prepareStatement(INSERT_EMPLOYEE_QUERY);){  //try2
                    //Set values to query
                    ps.setString(1,name);
                    ps.setString(2, addr);
                    ps.setDouble(3,Double.parseDouble(salary));
                    ps.setInt(4,Integer.parseInt(no));
                    //Execute the query
                    int count=ps.executeUpdate();
                    //process the results
                    if(count==0)
                        System.out.println("record not inserted");
                    else
                        System.out.println("Record Inserted");
                }//try2
            }//try1
        }catch(ClassNotFoundException cnf){
            cnf.printStackTrace();
            System.out.println("Problem in record insertion");
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}//class
```

```
Conclusion
    >> You can compare Manual Jobs, With Jobs, or other using Simple Employment Job
    >> You can Install CSS powered with Java conditions, JSON Data, another young Google Businesses Job
```

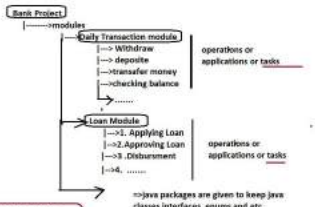
```
>> Utility Installations apps. Also.          >> Files with .txt and another data directory, by taking the columns as
Comma Separated Values.
>> Install Legal Options. this workspace uploaded files will not send for some Others. of this license and Email address
(after collected)** And they don't will be stored. this Will take, to Strong Islands. <br>and more. >>You prefer to  to access
ALWAYS FREE  Programmers and developers.
```

CallableStatement object

=>It is given to call PL/SQL procedures and functions of underlying DB s/w
=> Instead of writing common b.logic/ persistence logic in every module as java code or SQL Queries .. It is recomanded to write only 1 time in DB s/w as PL/SQL procedure or function and use it in multiple apps of a module or multiple modules of Project having reusability.

=> PL/SQL procedures /functions are developed in DB s/w , executed in Db s/w and also resides in Db s/w. So they are called Stored precedures and functions.

=> we use CallableStatement obj in multiple Apps of module or in multiple modules of a Project to call PL/SQL procedures and functions.



Project contains modules
module contains apps
app contains programs
programs contain instructions

=>java packages are given to keep java classes,interfaces ,enums and etc..
=>Modules and packages of the Project are logical seperations .. i.e they will not create any packages based seperation in project creation..
=>The persistence logic related classes different Apps or modules will be placed same package..

=> usecase for CallableStatement obj
a) Instead of writing login /authentication logic in every module .. It is recomanded to write only for 1 time as PL/SQL procedure /function and use it multiple modules
b) Instread of writing attendence calculation logic in every module ..of University Project .. it is recomanded to write only 1 time PL/SQL procedure /function and use it multiple modules.

In a s/w project (services project)
|---> 70% persistence logic (SQL queries)
|---> 30% persistence logic (PL/SQL procedures/Functions)

=>PL/SQL programing syntaxes are DB s/w dependent for 100%.

=>PL/SQL procedure does not return a value ... but we can use multiple out params to get multiple results from PL/SQL procedure
=>PL/SQL function return a value ... So we can use multiple out params and 1 return value to get multiple results from PL/SQL function.

If PL/SQL procedure wants to give 10 results then we need to take 10 out params
If PL/SQL function wants to give 10 results then we need to take 9 out params + 1 return value

=> The parameters of java method contains only type... where as the parameters of PL/SQL procedure or function contain both type and mode.

The possible modes are
IN :: INPUT param (default)
OUT :: OUTPUT param
INOUT :: INPUT and OUTPUT param

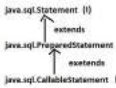
PL/SQL logic1:
z:=x+y;
z -->out param
x,y --> in params

In oracle PL/SQL programming
:= for assignment
== for comparision

PL/SQL logic2::
x:=x*x;
x --> INOUT params

Is

=>CallableStatement obj means it the object of java class given by underlying JDBC driver s/w implementing java.sql.CallableStatement(I).



=>CallableStatement object contains SQL query that is required to call PL/SQL procedure or function of Db s/w as pre-compiled SQL query.

Procedure to create PL/SQL procedure using SQL Plus at command prompt

step1) launch SQL Plus and authentication process.

```
SQL> connect
Enter user-name: system/manager
Connected.
```

Try to SQLCommand Line as window admin

step2) execute one SQL query

```
SQL> select * from student;
```

step3) Open "ed" editor write PL/SQL procedure by replacing the buffer query..

```
SQL> ed -->
create or replace procedure first_pro(x in number ,y in number ,z out number) as
begin
  z:=x+y;
end;
/
```

param name mode type

PL/SQL keywords and syntax is not case-sensitive

File menu -->save ---> file menu-->exit ---> SQL> /

Procedure created

(or)

note:: If we get Warning: Procedure created with compilation errors. like this then use SQL> show errors; to know the list of error messages..

```
4/1   PLS-00103: Encountered the symbol "END" when expecting one of the
      following:
      . ( * @ % & = - + ; < / > at in is mod remainder not rem
      <an exponent ("**")> <> or != or ~= >= <= <> and or like like2
      like4 likec between || member submultiset
      The symbol ";" was substituted for "END" to continue.
```

JDBC Data types

=>These are bridge data types b/w SQL Data types of Db s/w and java data types.
=>These data types gives instructions to JDBC driver s/w to convert java notation data to SQL notation data and vice-versa.

=> All JDBC data types are constants (public static final varible) of java.sql.Types class

java datatype	JDBC type	oracle SQL data type
int	Types.INTEGER	number
long	Types.LONG	number
java.lang.String	Types.VARCHAR	varchar,varchar2
java.sql.Date	Types.DATE	date
java.sql.Time	Types.TIME	time
byte[]	Types.BINARY	blob
char[]	Types.CHARACTER	clob
...	.....	...

Process/steps to fallow to call PL/SQL procedure from JAva App using CallableStatement obj

step1) create query calling PL/SQL procedure of DB s/w

```
String query="{call first_pro(?,?,?) }";
                  1 2 3
```

step2) create CallableStatement obj by making the query that calls PL/SQL procedure as the pre-compiled SQL query

```
CallableStatement cs=con.prepareCall(query);
```

sends the given Query to Db s/w to make it pre-compiled Query and returns CallableStatement obj having that pre-compiled query

step3) register OUT parameters of pre-compiled query with JDBC data types

```
cs.registerOutParameter(3,Types.INTEGER);
```

param index, JDBC type

note: All out parmeters of PL/SQL procedure or function must be registred with JDBC data types

step4) set values to IN parameters of Pre-compiled SQL query

```
cs.setInt( 1,1000);
c s.setInt(2,3000);
```

params index, value

=>sets values to IN parameters by interfly registering IN params with JDBC data types.

note:: there is no need of registering IN parameters with JDBC data types seperately.. they will be registered automatically when we call cs.setXxx(-,-) methods..

step 5) Call PL/SQL procedure

```
cs.execute();
```

note:: To execute/call same PL/SQL procedure for multiple times with different inputs then repeat step6 to step6 for multiple times

step6) gather results from OUT params using cs.getXxx(-) methods

```
int result=cs.getInt(3);
```

step7) close jdbc objs

```
cs.close();
con.close();
```

Example App on CallableStatement obj

```
//CsPRocedureTest.java
package com.nt.jdbc2;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Types;
import java.util.Scanner;

/*create or replace procedure first_proc(x in number ,y in number ,z out number) as
begin
 z:=x+y;
end;
*/
public class CsPRocedureTest {
   private static final String  CALL_PROCEDURE="{CALL FIRST_PROC(?,?,?) }";
      public static void main(String[] args) {
            //read inputs
           int first=0,second=0;
    try(Scanner sc=new Scanner(System.in)){
       if(sc!=null) {
       System.out.println("Enter first value::");
       first=sc.nextInt();
       System.out.println("Enter second value::");
       second=sc.nextInt();
       }
          //estblish the connection
       try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                    //create CallableStatemetn object having the query calling PL/SQL procedure as the pre-compiled SQL query
                    CallableStatement cs=con.prepareCall(CALL_PROCEDURE);){
                 //register OUT params with JDBC data types
                if(cs!=null)
                  cs.registerOutParameter(3,Types.INTEGER);
                //set values to IN params
                if(cs!=null) {
                    cs.setInt(1, first);
                    cs.setInt(2, second);
                }
                //execute/call thePL/SQL function
                if(cs!=null)
                    cs.execute();
                //gather results from OUT param
                int result=0;
                if(cs!=null)
                    result=cs.getInt(3);
                System.out.println("sum is ::"+result);
       }//try2
    }//try1
    catch(SQLException se) {
        se.printStackTrace();
    }
    catch(Exception e) {
        e.printStackTrace();
    }

      }//main
}//class
```

---

steps to follow to create PL/SQL procedure in oracle DB s/w using SQL Developer

step1) launch SQL developer and connect to oracle Db s/w

step2) **create PL/SQL procedure..**

```
CREATE OR REPLACE PROCEDURE P_GET_EMP_DETAILS_BY_ID
(
  NO IN NUMBER
, NAME OUT VARCHAR2
, DESG OUT VARCHAR2
, SALARY OUT VARCHAR2
) AS
BEGIN
                                    oup param names
  SELECT ENAME,JOB,SAL INTO NAME,DESG,SALARY from EMP  WHERE  EMPNO=NO;
            colum names
END;
```

we

when save the above code .. If noerrors there then it will be compiled otherwise errors will be displayed.

Example JDBC App , calling the above PL/SQL procedure

```
//CsPRocedureTest.java
package com.nt.jdbc2;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Types;
import java.util.Scanner;

/*CREATE OR REPLACE PROCEDURE P_GET_EMP_DETAILS_BY_ID
(
  NO IN NUMBER
, NAME OUT VARCHAR2
, DESG OUT VARCHAR2
, SALARY OUT VARCHAR2
) AS
BEGIN

  SELECT ENAME,JOB,SAL INTO NAME,DESG,SALARY from EMP  WHERE  EMPNO=NO;

END;
*/
public class CsPRocedureTest2 {
   private static final String  CALL_PROCEDURE="{CALL  P_GET_EMP_DETAILS_BY_ID(?,?,?,?) }";
      public static void main(String[] args) {
            //read inputs
            int eno=0;
    try(Scanner sc=new Scanner(System.in)){
       System.out.println(" enter employee details :: ");
       if(sc!=null) {
       System.out.println("Enter EmpNo::");
       eno=sc.nextInt();
       }
          //estblish the connection
       try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                    //create CallableStatemetn object having the query calling PL/SQL procedure as the pre-compiled SQL query
                    CallableStatement cs=con.prepareCall(CALL_PROCEDURE);){
                 //register OUT params with JDBC data types
                if(cs!=null) {
                      cs.registerOutParameter(2,Types.VARCHAR);
                      cs.registerOutParameter(3,Types.VARCHAR);
                      cs.registerOutParameter(4,Types.FLOAT);
                }

                //set values to IN paams
                if(cs!=null) {
                    cs.setInt(1, eno);
                }
                //execute/call thePL/SQL funciton
                if(cs!=null)
                    cs.execute();
                //gather results from OUT params .

                if(cs!=null) {
                   String name=cs.getString(2);
                   String desg=cs.getString(3);
                   float salary=cs.getFloat(4);
                   System.out.println("name::"+name+" desg="+desg+" salary="+salary);
                }

       }//try2
    }//try1
    catch(SQLException se) {
        System.out.println("requested data is not navalable");
        se.printStackTrace();
    }
    catch(Exception e) {
        e.printStackTrace();
    }

      }//main
}//class
```

Assignment :: Write JDBC App that calls PL/SQL precedure to get student name, addrs and avg based on given student number?

write A JDBC App that performs Authentication by calling PL/SQL procedure?

```
CREATE OR REPLACE PROCEDURE P_AUTHENTICATE
(
 USERNAME IN VARCHAR2
, PASSWORD IN VARCHAR2
, RESULT OUT VARCHAR2
) AS
  CNT NUMBER(5);            // It is Local variable
BEGIN
 SELECT COUNT(*) INTO CNT FROM IRCTC_TAB  WHERE UNAME=USERNAME AND PWD=PASSWORD;
 IF(CNT<>0) THEN
   RESULT:='VALID CREDENTIALS';
 ELSE
   RESULT:='INVALID CREDENTIALS';
 END IF;
END P_AUTHENTICATE;
```

```
   CsProcedureTest_Auth  .java
    package com.nt.jdbc2;

    import java.sql.CallableStatement;
    import java.sql.Connection;
    import java.sql.DriverManager;
    import java.sql.SQLException;
    import java.sql.Types;
    import java.util.Scanner;
    /*CREATE OR REPLACE PROCEDURE P_AUTHENTICATE
    (
     USERNAME IN VARCHAR2
    , PASSWORD IN VARCHAR2
    , RESULT OUT VARCHAR2
    ) AS
      CNT NUMBER(5);
    BEGIN
     SELECT COUNT(*) INTO CNT FROM IRCTC_TAB  WHERE UNAME=USERNAME AND PWD=PASSWORD;
     IF(CNT<>0) THEN
       RESULT:='VALID CREDENTIALS';
     ELSE
       RESULT:='INVALID CREDENTIALS';
     END IF;
    END P_AUTHENTICATE;
    */
    public class CsProcedureTest_Auth {
      private static final String CALL_PROCEDURE_QUERY="{CALL P_AUTHENTICATE(?,?,?)}";
          public static void main(String[] args) {
                //read inputs
                try(Scanner sc=new Scanner(System.in)){
                     String username=null,password=null;
                      if(sc!=null) {
                            System.out.println("enter username::");
                            username=sc.next();
                            System.out.println("Enter password::");
                            password=sc.next();
                      }
                      try{  //establish the connection
                                Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                            //create CallableStatement obj
                                    CallableStatement cs=con.prepareCall(CALL_PROCEDURE_QUERY);{

                            //register OUT parameter with JDBC types
                                if(cs!=null)
                                  cs.registerOutParameter(3,Types.VARCHAR);

                               //set values to IN params
                                 if(cs!=null) {
                                   cs.setString(1,username);
                                   cs.setString(2, password);
                                 }
                               //call PL/SQL procedure
                                 if(cs!=null)
                                     cs.execute();
                               //get results from OUT param
                               String result=null;
                               if(cs!=null)
                                   result=cs.getString(3);
                               System.out.println(result);
                                   }//try2
                           }//try1
                catch(SQLException se) {
                      se.printStackTrace();
                }
                catch(Exception e) {
                      e.printStackTrace();
                }

          }//main
    }//class
```

=>The SQL Query that is there in CallableStatement obj is pre-compiled SQL query.. So there is no SQL injection problem even with CallableStatement obj.

---

number, varchar, varchar2, date and etc.. type params and variable can not 1 or more records given SQL select Query execution.. For that we can use Cursor variable.

=> A Cursor is a inMemory variable of oracle PL/SQL programming that can hold bunch of records (1 or more) given SQLselect Query execution..

SYS_REFCURSOR is built-in cursor data type in oracle PL/SQL programming whose variable can store multiple records

details SYS_REFCURSOR; ⟶ Cursor variable decl

open details for => opening the cursor
SELECT * FROM STUDENT WHERE SADD='hyd'

The records generated by SQL select Query will be stored in "details" variable (cursor variable)

PL/SQL programming "cursor" is equal to JDBC ResultSet(To hold bunch of record) in Java programming

number, varchar, varchar2, date and etc.. type params and variable can not 1 or more records given SQL select Query exeution.. For that we can use Cursor variable.
=> A Cursor is a InMemory variable of oracle PL/SQL programming that can hold bunch of records (1 or more) given SQLselect Query execution..

SYS_REFCURSOR is built-in cursor data type in oracle PL/SQL programming whose variable can store multiple records

details SYS_REFCURSOR; ⟶ Cursor variable decl

open details for --> opening the cursor
SELECT * FROM STUDENT WHERE SADD='hyd'

| The records generated by SQL select Query will be stored in "details" variable (cursor variable)

PL/SQL programming "cursor" is equal to JDBC ResultSet(To hold bunch of record) in Java programming

=>There is not JDBC types in java.sql.Types class to register Cursor type OUT parameter with JDBC types.. So we need to use Oracle DB s/w specific OracleTypes.CURSOR for the same.
(part of ojdbc6/7/8.jar file)

```
cs.registerOutParameter(_,OracleTypes.CURSOR);
```

=>To gather Results from CURSOR OUT parameter after executing PL/SQL procedure use

```
ResultSet rs=(ResultSet)cs.getObject(-);
while(rs.next()){
   ....
   ....
}
```

Write a JDBC App that calls PL/SQL proecure to get emp details based on the given initial charcters of emp name

```
CREATE OR REPLACE PROCEDURE P_GET_EMPS_BYNAME_INITIAL
(
 INITIALCHARS IN VARCHAR2
, DETAILS OUT SYS_REFCURSOR
) AS
BEGIN

 OPEN DETAILS FOR
   SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE ENAME LIKE INITIALCHARS;

END P_GET_EMPS_BYNAME_INITIAL;
```

The records given by SELECT SQL Query will be stored into this "details" CURSOR

```
public class CsProcedureCursorTest {
 private static final String PROCEDURE_CALL_QUERY="{CALL P_GET_EMPS_BYNAME_INITIAL(?,?)} ";
    public static void main(String[] args) {

      try(Scanner sc=new Scanner(System.in)) {
         String initChars=null;
         if(sc!=null) {
            System.out.println("Enter initial chars of employee name::");
            initChars=sc.next()+"%";
         }

         try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
                 CallableStatement cs=con.prepareCall(PROCEDURE_CALL_QUERY);
                 ){
            //register OUT parameter with JDBC type
            if(cs!=null)
               cs.registerOutParameter(2, OracleTypes.CURSOR);
            //set value to IN param
            if(cs!=null)
               cs.setString(1,initChars);
            //execute or call PL/SQL procedure
            if(cs!=null)
               cs.execute();
            //gather results from OUT parameters
            if(cs!=null) {
               ResultSet rs=(ResultSet)cs.getObject(2);
               System.out.println("The output is");
               boolean flag=false;
               while(rs.next()) {
                  flag=true;
                  System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4)+" "+rs.getInt(5));
               }//while
               if(flag==false)
                  System.out.println("Records not found");
            }//if
         }//try2
      }//try1
      catch(SQLException se) {
         se.printStackTrace();
      }
      catch(Exception e) {
         e.printStackTrace();
      }

    }//main
}//class
```

=>PL/SQL function returns a value i.e If u want to get 10 output from PL/SQL function we must take one output as the return value and remainig outputs as the OUT parameters.

=>The SQL92 Query that calls PL/SQL function must one additional param reprsenting return value.. and this parameter should be treated as OUT parameter and must be registered with JDBC type.

These syntaxes are called SQL 92 syntaxes

syntax for calling PL/SQL function
{?=call <function-name>(?,?,?,....)}
1 2,3,4,....

| The return parameter index is "1" always..

syntax for calling PL/SQL procedure
{ CALL procedureName>(?,?,?,....)}

Write a JDBC App that calls PL/SQL function to get Student details based on the given student no?

```
CREATE OR REPLACE FUNCTION FX_GET_STUDENT_DETAILS_BY_NO
(
 NO IN NUMBER
, NAME OUT VARCHAR2
, ADDRS OUT VARCHAR2
) RETURN FLOAT AS          ---> Return type
 PERCENTAGE FLOAT;              ---> Local variable
BEGIN

 SELECT SNAME,SADD,AVG INTO NAME,ADDRS,PERCENTAGE FROM STUDENT WHERE SNO=NO;
           col names            out params   Local variable                 col Name  in param
 RETURN PERCENTAGE;
END FX_GET_STUDENT_DETAILS_BY_NO;
```

```
package com.nt.jdbc2;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Types;
import java.util.Scanner;

/*CREATE OR REPLACE FUNCTION FX_GET_STUDENT_DETAILS_BY_NO
(
 NO IN NUMBER
, NAME OUT VARCHAR2
, ADDRS OUT VARCHAR2
) RETURN FLOAT AS
 PERCENTAGE FLOAT;
BEGIN

 SELECT SNAME,SADD,AVG INTO NAME,ADDRS,PERCENTAGE FROM STUDENT WHERE SNO=NO;

 RETURN PERCENTAGE;
END FX_GET_STUDENT_DETAILS_BY_NO;
*/
public class CsFunctionTest {
  private static final String CALL_FX_QUERY="{?=call FX_GET_STUDENT_DETAILS_BY_NO(?,?,?) }";
     public static void main(String[] args) {

          try(Scanner sc=new Scanner(System.in)){
               //read inputs
               int no=0;
               if(sc!=null) {
                    System.out.println("Enter Student number::");
                    no=sc.nextInt();
               }
               try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                         CallableStatement cs=con.prepareCall(CALL_FX_QUERY);
                         ){
                    //register return,OUT params with JDBC types
                    if(cs!=null) {
                         cs.registerOutParameter(1,Types.FLOAT); //return param
                         cs.registerOutParameter(3,Types.VARCHAR); //outparameter (index 3)
                         cs.registerOutParameter(4,Types.VARCHAR); //outparameter (index 4)
                    }
                    //set values to IN params
                    if(cs!=null)
                         cs.setInt(2, no);
                    //call/execute PL/SQL function
                    if(cs!=null)
                         cs.execute();
                    // gather results from return,OUT params
                    if(cs!=null) {
                         System.out.println("student name:: "+cs.getString(3)); //out param
                         System.out.println("Student addrs::"+cs.getString(4)); //out param
                         System.out.println("Student avg::"+cs.getFloat(1)); //return param
                    }
               }//try2
          }//try1
          catch(SQLException se) {
               System.out.println("Records not found");
               se.printStackTrace();
          }
          catch(Exception e) {
               e.printStackTrace();
          }

     }//main
}//class
```

Q) Write a JDBC App that calls PL/SQL procedure of MYSQL givens all records of Student db table based on the given addrs

Srpring Course
spring boot and MS
(If u have time do both courses)
(3500+ 5000)
100+ 100 sessions

spring basics, Spring boot and MS
(If u have less time)
⇩
(6pm slot july3rd week)
(rs :: 5000)
(120+ sessions)

Conclusion of statement objs
==========================
a) we can use all 3 statement objs in application
b) use simple statement to execute SQL query only for 1 time that to with out input values
c) use Prepared Statement to execute SQL query for times with or with out conditions, To execute SQL query with condition only for 1 time or multiple times , to insert date values, LOBS(files) and etc..
d)Use CallableStatement obj to call PL/SQL proedures and functions of underlying DB s/w

write A JDBC App that performs Authentication by calling PL/SQL procedure?

```
CREATE OR REPLACE PROCEDURE P_AUTHENTICATE
(
 USERNAME IN VARCHAR2
, PASSWORD IN VARCHAR2
, RESULT OUT VARCHAR2
) AS
  CNT NUMBER(5);            // It is Local variable
BEGIN
 SELECT COUNT(*) INTO CNT FROM IRCTC_TAB  WHERE UNAME=USERNAME AND PWD=PASSWORD;
 IF(CNT<>0) THEN
   RESULT:='VALID CREDENTIALS';
 ELSE
   RESULT:='INVALID CREDENTIALS';
 END IF;
END P_AUTHENTICATE;
```

```
   CsProcedureTest_Auth  .java
    package com.nt.jdbc2;

    import java.sql.CallableStatement;
    import java.sql.Connection;
    import java.sql.DriverManager;
    import java.sql.SQLException;
    import java.sql.Types;
    import java.util.Scanner;
    /*CREATE OR REPLACE PROCEDURE P_AUTHENTICATE
    (
     USERNAME IN VARCHAR2
    , PASSWORD IN VARCHAR2
    , RESULT OUT VARCHAR2
    ) AS
      CNT NUMBER(5);
    BEGIN
     SELECT COUNT(*) INTO CNT FROM IRCTC_TAB  WHERE UNAME=USERNAME AND PWD=PASSWORD;
     IF(CNT<>0) THEN
       RESULT:='VALID CREDENTIALS';
     ELSE
       RESULT:='INVALID CREDENTIALS';
     END IF;
    END P_AUTHENTICATE;
    */
    public class CsProcedureTest_Auth {
      private static final String CALL_PROCEDURE_QUERY="{CALL P_AUTHENTICATE(?,?,?)}";
          public static void main(String[] args) {
                //read inputs
                try(Scanner sc=new Scanner(System.in)){
                     String username=null,password=null;
                      if(sc!=null) {
                            System.out.println("enter username::");
                            username=sc.next();
                            System.out.println("Enter password::");
                            password=sc.next();
                      }
                      try{  //establish the connection
                                Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
                            //create CallableStatement obj
                                    CallableStatement cs=con.prepareCall(CALL_PROCEDURE_QUERY);{

                            //register OUT parameter with JDBC types
                                if(cs!=null)
                                  cs.registerOutParameter(3,Types.VARCHAR);

                               //set values to IN params
                                 if(cs!=null) {
                                   cs.setString(1,username);
                                   cs.setString(2, password);
                                 }
                               //call PL/SQL procedure
                                 if(cs!=null)
                                     cs.execute();
                               //get results from OUT param
                               String result=null;
                               if(cs!=null)
                                   result=cs.getString(3);
                               System.out.println(result);
                                   }//try2
                           }//try1
                catch(SQLException se) {
                      se.printStackTrace();
                }
                catch(Exception e) {
                      e.printStackTrace();
                }

          }//main
    }//class
```

=>The SQL Query that is there in CallableStatement obj is pre-compiled SQL query.. So there is no SQL injection problem even with CallableStatement obj.

---

number, varchar, varchar2, date and etc.. type params and variable can not 1 or more records given SQL select Query execution.. For that we can use Cursor variable.

=> A Cursor is a inMemory variable of oracle PL/SQL programming that can hold bunch of records (1 or more) given SQLselect Query execution..

SYS_REFCURSOR is built-in cursor data type in oracle PL/SQL programming whose variable can store multiple records

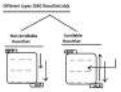
details SYS_REFCURSOR; ⟶ Cursor variable decl

open details for => opening the cursor
SELECT * FROM STUDENT WHERE SADD='hyd'

The records generated by SQL select Query will be stored in "details" variable (cursor variable)

PL/SQL programming "cursor" is equal to JDBC ResultSet(To hold bunch of record) in Java programming

```
OffStreet Layer, DAC ResultSet.club
```



```
Non-scrollable
ResultSet
```



```
Scrollable
ResultSet
```

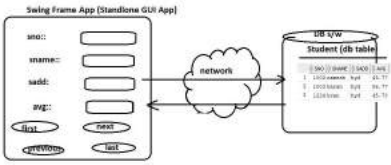


```
Non-scrollable RS
——————————
-- The 'fill' from either no-arrow records will throw/modify/drop in-bottom1, unpossibly  is called
new scrollable RS
-- RS default name RS non-scrollable RS

   ResultSet rs=con.createStatement();
   ResultSet rs=st.executeQuery("SELECT * FROM TAYLOR");
            rs(n) rs.next();
Scrollable RS
~~~~~~~~~~~~~~
-- The 'fill' from allows us to access the records' non-sequentially (it) either/locally , randomly, already is called
   scrollable RS
-- RS default name 'rs' statement obj having type,mode,  then that statement obj allows us to create
   rs scrollable RS

-- Both shapin statement, PreparedStatement support to  non-scrollable and scrollable RS

  To create scrollable using  1Single statement obj
  ——————————————————————————————————————————————
   Statement stmt=con.createStatement(type,mode);
   ResultSet rs=stmt.executeQuery("SELECT * FROM TAYLOR");
            scrollable RS

RS create for/editable RS using PreparedStatement obj
——————————————————————————————————————————————
PreparedStatement pst=
    con.prepareStatement("SELECT * FROM RS NAME ",
                         ResultSet.
                                 TYPE,
                         ResultSet._mode rs=pst.executeQuery();
                             con.scrollable RS

-- The con/from scrollable or non-scrollable RS
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    rs.next()
    rs.last()
    rs.getRow()
    rs.getRow() : retrieve the current position of the cursor in this RS
                                                  position
                                                  number
                                                  of the row
                                                                 the any-returns PL/SQL programming cursor or
                                                                 cursor cursor in 'rs.last.pos()  cursor')

Non-scrollable that can be movable on to scrollable RS
——————————————————————————————————————————————
    rs.next()
    rs.previous()
    rs.absolute()
    rs.first()
    rs.last()
    rs.beforeFirst()
    rs.afterLast()
    rs.isFirst()
    rs.isLast()
    rs.getRow() : ——————RS pointer, whether cursor is there in 'first record' or not
    rs.last()
    rs.moveToInsertRow()
    rs.deleteRow()
    rs.absolute(-1)  Moves the cursor to RS either in forward or in backward direction.
    rs.absolute(1) or
      rs.next()...
                        where is the difference to 'rs.relative(1)' or(1) and absolute(1)'=1?...
                                    ~~~~~~~~~~~~~~~~~~~
```

=>In java stadalone Apps .. to add GUIness we can take the support AWT/swing,
=> In web applications the GUIness comes becoz html tags and form comps.

=>AWT is outdated .. becoz it is not giving more comps to work with
(AWT :: Abstract window Tool kit)
=>Swing is built on the top of AWT .. more over it given mutliple basic and advanced comps to work with

=>Java App + JDBC Code +AWT/Swing will be treated as GUI Front end to the Back end called DB s/w



There is no next record for last record , So before calling rs.next() method ensure that the record pointer is not already there in last record by using if(!rs.isLast()) method.

There is no previous record for first record , So before calling rs.previous() method ensure that the record pointer is not already there in first record by using if(!rs.isFirst()) method.

Container :: The comp /place where other GUI comps will be placed is called container
---
JFrame

comps

JLabel --->4
JTextField -->4
JButton --->4

LayoutManager (Specifies how comps should be arranged)

FlowLayout (Horizontally central)

Event

ActionEvent (c)
(raises when the comps are clicked)
(Button)

EventListener

ActionListener (I)

event handling method ---------- Method decl in ActionListener(I)
---
public void actionPerformed(ActionEvent e){
....
}

Different Containers ::::
JFrame :: frame for standalone Apps
JPanel :: for sub frame
JApplet :: java based web page
JDialogBox :: Torender dialog box
JCanvas :: for drawings..

Swing/AWT containers are not releated to SErvletcontainer,jsp container ,spring's IOC container

other layout managers are

=>BorderLayout
=>GridLayout
=>CustomLayout/NullLayout
and etc..

=>The action performed on the comp is called Event..executing some logic for event that is raised called Event handling.. EventListener are resposible providing Event handling metods where write the event handling logic.

note:: we do not call event handling methods manually becoz they will be called automatically the moment event is raised ,so the event handling methods are also called as callback methods.

note:: when click on Button , the action event will be raised.. To handle that event we take ActionListener(I) impl class by placing event handling logic in the event handling method "public void actionPerformed(-)"

=>To make ur class a JFrame window/container , extends it from javax.swing.JFrame(c)
=>To make ur class a Event Listener , extends it from EventListener(I) directly or indirectly
=>To make ur class a ActionEvent Listener , extends it from ActionListener(I) directly or indirectly and write the event handling logic using "public void actionPerformed(-) method".

=> same class methods , super class public class methods in sub class can be called directly with out object or reference variable.

>>in standalone CLI Apps , we try to create only connection with DB a fw for all persistence operations since only one user will operate the app on a time. but the logics to make connections are localized. in webs static block or connection that executes only for 1 time. >>entrances is same basic >> in webs , we have static and non-static variables in the controllers , where as in static like in war can only static variables >> same class methods and maps public methods in sub class can be called directly with out using any object or refer variable.

>>if u want to use logics certain methods only on the current class having modifiy then write calling those methods as private methods

>> members method() is given to write the data in text box and members getData() is given to read the data from that box.

Example Code

```
package com.vt.dbc;

import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.awt.event.WindowStateListener;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.SQLSyntaxErrorException;
import java.sql.Statement;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class DbcUtilCreateTableUtil extends JFrame implements ActionListener,WindowListener {
    private static final String  GET_STUDENTS_CLOB="SELECT SID,SNAME,MARKS,ADDRS FROM STUDENT";
    private JLabel lno,lname,ladd,lmarks;
    private JTextField tno,tname,tadd,tmarks;
    private JButton  btnGetStudentsClob;
    private Connection con;
    private PreparedStatement ps;
    private  ResultSet rs;

    @OverRide
    public CLOBJustFrameTest() {
        System.out.println("CLOBJustFrameTest::0-param constructor");
        setTitle("CLOB Read Write Example");
        setSize(400,400);
        setLayout(null);

        //add comps
        lno=new JLabel("sno");
        add(lno);
        tno=new JTextField(10);
        add(tno);

        lname=new JLabel("name");
        add(lname);
        tname=new JTextField(10);
        add(tname);

        ladd=new JLabel("addr");
        add(ladd);
        tadd=new JTextField(10);
        add(tadd);

        lmarks=new JLabel("marks");
        add(lmarks);
        tmarks=new JTextField(10);
        add(tmarks);


        btnInsert=new JButton("Insert");
        add(btnInsert);
        btnInsert.addActionListener(this);
        btnUpdate=new JButton("Update");
        add(btnUpdate);
        btnUpdate.addActionListener(this);
        btnDelete=new JButton("Delete");
        add(btnDelete);
        btnDelete.addActionListener(this);
        btnGetStudentsClob=new JButton("GetStudentsClob");
        add(btnGetStudentsClob);
        btnGetStudentsClob.addActionListener(this);

        addWindowListener(this);
        setVisible(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);                  // Use closing of  frame window will terminate Application .
        initializeDB();
        System.out.println("CLOBJustFrameTest::0-param constructor End");
    }

    private  void  initializeDB(){
        System.out.println("CLOBJustFrameTest.initializeDB()");
        try {
            //1.load the driver class
            Class.forName("oracle.jdbc.driver.OracleDriver");
            //2.create Connection object
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
            //3.create PreparedStatement obj
            ps=con.prepareStatement("SELECT_CLOBDATA_CLOB_FROM_STUDENT_CLOB_DATA",
                                    ResultSet.TYPE_SCROLL_SENSITIVE,
                                    ResultSet.CONCUR_UPDATABLE);

            //4.prepare to execute PS obj
            rs=ps.executeQuery();
        }catch(SQLException se){
            se.printStackTrace();
        }catch(Exception e){
            e.printStackTrace();
        }
    }//initializeDB()

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            new CLOBJustFrameTest();  //anonymous class
            System.out.println("end of main method");
        });
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        System.out.println("actionListener is fired");
        String btnLabel=ae.getActionCommand();
        if(btnLabel.equals("Insert")){
            try {
                int[] ir=null;
                Flag=true;
                if(flag){
                    System.out.println("first button is clicked");

                    ps.addBatch();
                    ir=ps.executeBatch();
                }
            }catch(BatchUpdateException be) {
                be.printStackTrace();
            }catch(SQLException se) {
                se.printStackTrace();
            }
        }else if(btnLabel.equals("Update")){
            System.out.println("next button is clicked");
            try {
                int[] ir=null;
                ps.clearBatch();
                Flag=true;
                while(rs.next()){
                    flag=false;
                }
                if(flag){
                    System.out.println("no records found to update");
                }else{
                    ps.addBatch();
                    ir=ps.executeBatch();
                }
            }catch(BatchUpdateException be) {
                be.printStackTrace();
            }catch(SQLException se) {
                se.printStackTrace();
            }
        }else if(btnLabel.equals("Delete")){
            System.out.println("last button is clicked");
            try {
                ps.clearBatch();
                flag=true;
                while(rs.next()){
                    flag=false;
                }
                if(flag){
                    System.out.println("no records found to delete");
                }else{
                    ps.addBatch();
                    ir=ps.executeBatch();
                }
            }catch(BatchUpdateException be) {
                be.printStackTrace();
            }catch(SQLException se) {
                se.printStackTrace();
            }
        }else if(btnLabel.equals("GetStudentsClob")){
            try {
                String sno=tno.getText();
                int no=Integer.parseInt(sno);
                ps.setInt(1,no);
                ResultSet rs=ps.executeQuery();
                if(rs.next()){
                    Clob clob=rs.getClob(2);
                    BufferedReader br=new BufferedReader(clob.getCharacterStream());
                    String str=null;
                    while((str=br.readLine())!=null){
                        System.out.println(str);
                    }
                }else{
                    System.out.println("no records found");
                }
            }catch(SQLException se) {
                se.printStackTrace();
            }catch(Exception e) {
                e.printStackTrace();
            }
        }
    }//actionPerformed

    @Override
    public void windowOpened(WindowEvent e) {
        // TODO Auto-generated method stub

    }

    @Override
    public void windowClosing(WindowEvent e) {
        System.out.println("WindowEventHandler.windowClosing()");
        try {
            ps.close();
            con.close();
        }catch(SQLException se) {
            se.printStackTrace();
        }catch(Exception e) {
            e.printStackTrace();
        }
        try {
            System.exit(0);
        }catch(SecurityException se) {
            se.printStackTrace();
        }
    }

    @Override
    public void windowClosed(WindowEvent e) {
        // TODO Auto-generated method stub

    }

    @Override
    public void windowIconified(WindowEvent e) {
        // TODO Auto-generated method stub

    }

    @Override
    public void windowDeiconified(WindowEvent e) {
        // TODO Auto-generated method stub

    }

    @Override
    public void windowActivated(WindowEvent e) {
        // TODO Auto-generated method stub

    }

    @Override
    public void windowDeactivated(WindowEvent e) {
        // TODO Auto-generated method stub

    }

}//class
```

How to make standalone GUI Java app in the Desktop App?

Ans) Use Executable JAR file... [Jarwiz is a built-in tool of jdk 1.6x]

--->If a c++ GUI App is developed using Eclipse IDE, then make sure that the exe file will only run when and if Eclipse Project IDE are same... atleast the installer java version must smaller than eclipse Project base version

note:- To change eclipse Java version of installed IDE - just Shut (restart) ECLIPSE_HOME\ Ini\ directory

note:- To change java version in eclipse base Project, right click on project ---> buildpath ---> My Workspace ---> libraries tag ---> edit library ---> left upmost directory ---> installed Jres ---> add ---> installed Vm ---> browse ---> select Java version dir. (Inserting JDK Installation Folder ---> ... ---> ...)

Working with GUI Tool

Right click on desktop ---> new ---> shortcut --->

Note :- before next Rich suffix NOTE.txt file is added incomplete text... variable CLASSPATH.

double click on exe file

select Jrunner ---> browse ---> ... ---> select Jar file

give shortcut name ---> [C:\Users\Chandra\Desktop\Jarwiz.exe]

Next Screen of the above package save as jar.bat

---> In eclipse workspace, GUI serial Repository folder the exe folder + shortcut bat file + source code (.class files)... and .bat directory available... else of .exe files

---> Jrunner, Jarwiz, App (optional), ---> change icon (optional) ---> apply ---> ok

Insert the app by double clicking on the icon

How can we make GUI Java app as the desktop App?

Ans) Take the support .bat (batch) file into ...

run bat (create in .exe windows)

---

ref. C:\Users\Chandra\Desktop\Jarwiz.exe

java -cp C:\Users\Chandra\Desktop\Jarwiz.exe; C:\Users\Chandra\Desktop\Jarwiz.exe; C:\Users\Chandra\Desktop\Jarwiz.exe;

Procedure convert to.java init log error as new file

new app ——> jar file (executable jar file) ——————> exe file

steps[ ]
Executable jar file having manifest file representing the .exe app

F:\Workshops
|——>bridgelabz
|——>MODULE
|——>NewFile
|——>impressed.inf
|——>Audio call kannal.wav.java 1.0.0 Ajmal K ©2024 JustWrote.java
|——>Use_of_Jdbc3
|——>MSTeamsBrowserHelp1.class

impressed.inf (InstallInfo file) (key: filename= exec can be taken)

Manifest.txt(exe) : 1.0 ✔
Main-class : com.in.jdbc3.batchmidleproject ✔

F:\Workshops\JustWrote\MSTEAMS\Classes>jar cvmf impressed.inf mypp1.jar
|——>META-INF\MANIFEST.MF
|——>com\Main$1.class
|——>com\Main.class
|——>com\Main.java
Everything of current directory will placed in an mypp1.jar file by taking important info as the manifest file.

Note:- If u r placing on external .exe& mypp1.jar file ... then the manifest file must consider the main class name that should be used to start the execution.

steps[] Download IoT Tool (via Eclipse Installer) from Internet

Download from https://www.eclipse.org/downloads/ ——> in zip file (jdx_x86 zip) ——>
extract zip file to get setup ——> install setup file

steps[] use .exe installed tool(Internet or network) as file to exe file.

Result ——>

steps[] Run the Application

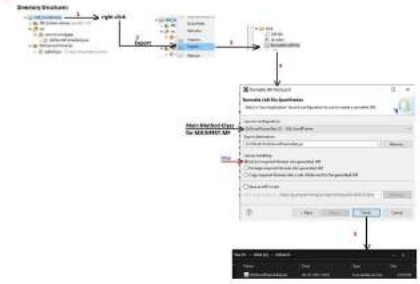
Executable Jar to .exe Conversion Using "Jar2exe" S/W

Devesh Pandey
devpandey16@gmail.com

1> Create executable jar using command prompt.
OR
Using Eclipse-Gui-tool (Recommended)
Because eclipse gives the option to extract required file into generated jar file so we need not do place explicitly DEMO.jar

Ex:

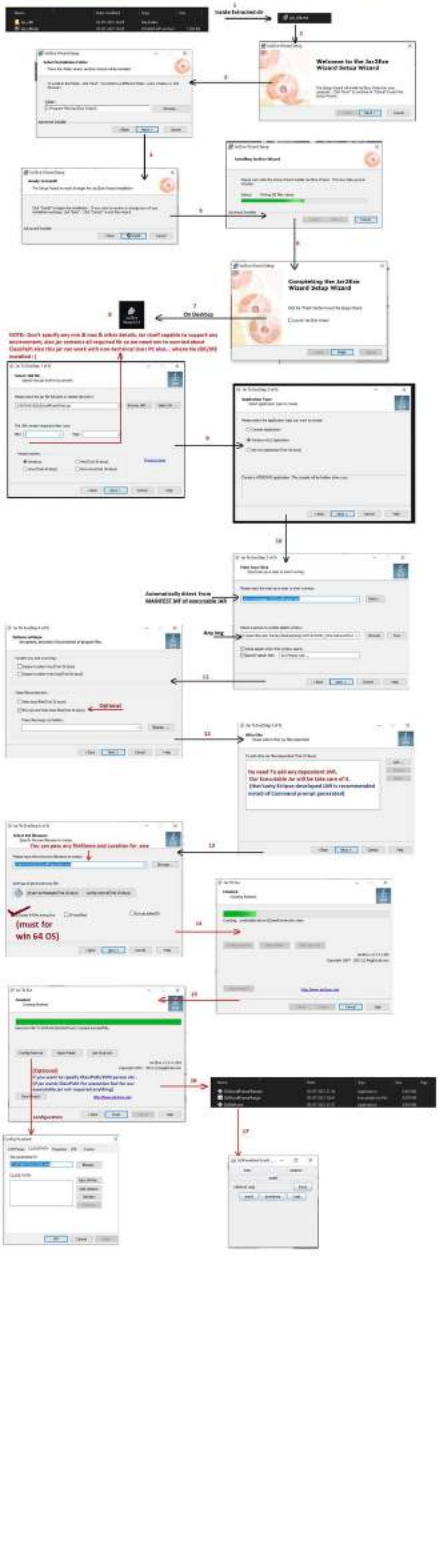
Directory Structure:

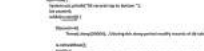


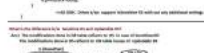
2> Download and Install "Jar2Exe" S/W.

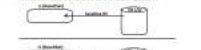
Devesh Pandey
devpandey16@gmail.com

Links: https://www.jar2exe.com/downloads









There are 4+1 mechanims /architectures /methodologies to develop JDBC driver s/ws for different Db s/ws based on the rules and guidelines of JDBC Technology.

1.Type1 JDBC driver (JDBC-ODBC Bridge Driver) | These are official architectures given by Sun Ms ..
2.Type2 JDBC driver (Native API /Partly JAva Driver) |
3.Type3 JDBC driver (NetProtocol /All JAva Driver) |
4.Type4 JDBC driver (NativeProtocol /All JAva Driver) |
5.Type5 JDBC driver (Not official :: No Technical name) | Unofficial architecture given by Sun Ms.

=> All ODBC driver s/ws and some JDBC driver s/ws locate and interact with Db s/w by using Vendor DB Library or Native DB interface. Generally this comes along with Db s/w installation as exe file or dll file .. In case of oracle the Vendor DB library is oci.dll.

oci --->oracle call interface.

dll -> dynamic linking library (advanced exe file)

=> Applets are compiled java classes .. which can be sent over the network as java based web pages..

Two types of applets (out dated)

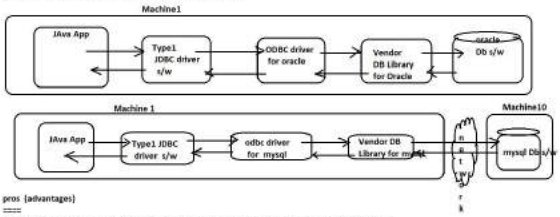
a) Untrusted Appets (By default all applets are untrusted Applets)
(These applets can not with Interact with file System of the underlying computer ...
So they can not write virus to the computer. These are safe applets)

b) Trusted Appets
(These applets can with interact with file System of the underlying computer ...
So they can write virus to the computer.. These are not safe applets.)

Type1 JDBC driver architecture

=>Type1 JDBC driver s/ws are not given to interact with Db s/ws directly.. They are given to locate and interact with DB s/ws though ODBC drivers and Vendor DB Libraries..

=>Since Java Apps can not ODBC drivers directly becoz their pointers utilization..So Type1 JDBC driver s/w is develop to act as bridge JAva App and ODBC driver..



pros (advantages)

=> One type1 JDBC driver s/w is sufficient to interact and interact with multiple Db s/ws by using mutliple ODBC drivers and VendorDb libaries
note:: Vendor DBLibraries and ODBC drivers are Db s/w specific and they are 1 set per each DB s/w
=> Upto Java8 , This type1 JDBC driver is built-in with jdk s/w.
=>suitable for testing of projects..

cons(disadvantages)

=> Since multiple comps are involved before interacting with Db s/w.. This Driver performence very poor
=>Not suitable medium scale and Large scale projects
=>Since Vendor DB Library and ODBC driver are required in the Client machine so not suitable for Unstrusted Applets to Db s/w communication
=> All methods in JDBC driver s/w classes are synchronized methods.. i.e this driver s/w thread safe by default and degrades the performecne
=>Only Sun Ms is giving type1 jdbc driver s/w .. that to upto java8 version.. after that version we do not have type1 JDBC driver s/w
=> Not a industry standard JDBC driver s/w.

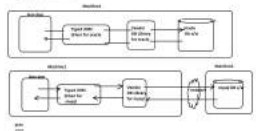
Basic, as in 2 interfaces (architecture) / bothknowledge: to develop ANY driver (also for different OS) is be based on the rules and guidelines of ANSI Technology

1.Type1 ODBC driver (ODBC-ODBC Bridge Driver)
2.Type2 JDBC driver (Native API Partly Java driver)
3.Type3 JDBC driver (RadPartProtocol (All Java Driver)
4.Type4 JDBC driver (NativeProtocol (All Java Driver)
5.Type5 JDBC Driver (2nd official) - for Technical server

There are different architectures where its live fit...

Unofficial architecture given by me) like...

4.Type4 JDBC driver (Native API Partly Java driver)

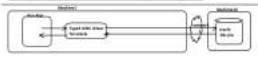


pros
cons
=> No need of arranging JDBC drivers
=> Gives significantly better performance compare to type1 jdbc driver

cons
pros
=>Pure Vendor DB Library is required... This driver is not suitable somewhere hopkins to DB s/w communication
=> This driver performance is assigned for medium and large program
=> This must be arranged seperately from DB vendors or Third party vendors
=> It is many DB s/w , we need seperate Type4 jdbc driver s/w
Exp: Type4 JDBC drivers s/w For oracle , type4 JDBC drivers s/w for mysql)

3.Type3 JDBC driver (RadPartProtocol (All Java Driver)

=>This jdbc driver locates Java DB s/w directly with out using any Vendor DB Library and ODBC driver.
=>This driver is usually developed Java DB Java driver.
=>This driver uses underlying DB s/w specific logics (neither protocol nor locals are known with that DB s/w.



pros
cons
=> No need of arranging jdbc drivers and vendor DB Libraries
=> This driver gives performance compare to type1,type2 drivers
=> Pure driver size is suitable for modern webs and large scale projects
=> Since ODBC drivers and Vendor DB Library are not required , one can mix this driver for w/s/m/cloud Applic to DB s/w communication
=> Since this driver is developed in pure java, it is platform independent jdbc driver
=> This driver can be downloaded as an .exe program dynamically from Internet using reasonegrades exam.

cons
=>we must arrange there JDBC drivers seperately from DB vendors or Third party vendors
=>If we many DB s/w, we must seperate type4 JDBC driver s/w.
Exp: type4 jdbc driver for oracle , type4 jdbc driver for mysql and soon...)

Type3 JDBC Driver

Socket Programming

Some this support for reflection to implement this()

Client

Server

Client Server Socket (It works in local) (It works in Internet) (It works in Connectivity) of same thread

Socket

Socket

Socket

=>Oracle Corp supplied Type2 jdbc driver for oracle is called oci driver and type4 jdbc driver oracle is called oracle thin driver.. Both these driver s/w are coming in single jar file called ojdbc<n>.jar.

oci ::oracle call interface.

oci driver details
==========
model : type2 JDBC driver
target Db s/w :: oracle
driver class name : oracle.jdbc.driver.OracleDriver
jdbc url :: jdbc:oracle:oci8:@<sid>
protocol sub name logical db name
jar file :: ojdbc<n>.jar

all these details are same as oracle thin driver(type4 driver) expect jdbc url .

=>Based on the sub name in jdbc url, the oracle thin driver or oci driver will be pickedup
If the sub name is "oci8" then type2 jdbc driver (oci driver)
If the sub name is "thin" then type4 jdbc driver (thi driver)

Example App
=========
=>Take any oracle thin driver app and change the url ...

```
public class PSLoginApp_Type2_OCI {
  private static final String LOGIN_QUERY="SELECT COUNT(*) FROM IRCTC_TAB WHERE UNAME=? AND PWD=?";
    public static void main(String[] args) {
        Scanner sc=null;
        Connection con=null;
        PreparedStatement ps=null;
        ResultSet rs=null;
        try {
            //read inputs
            sc=new Scanner(System.in);
            String user=null,pass=null;

            if(sc!=null) {
                System.out.println("enter Login username::");
                user=sc.nextLine(); //gives raja rao' --
                System.out.println("enter Login password::");
                pass=sc.nextLine(); //gives rao rao
            }//if

            //Load JDBc driver class (optional)
            Class.forName("oracle.jdbc.driver.OracleDriver");
            //establish the connection
            con=DriverManager.getConnection("jdbc:oracle:oci8:@xe", "system","manager");
            //create JDBC PreparedStatement object
            if(con!=null)
                ps=con.prepareStatement(LOGIN_QUERY);

            //set values to the params of pre-compiled SQL query
            if(ps!=null) {
                ps.setString(1,user);
                ps.setString(2, pass);
            }

             //send and execute the SQL query in Db s/w
             if(ps!=null)
                 rs=ps.executeQuery();

             //process the ResultSet object
             if(rs!=null) {
                 rs.next(); //moves the cursor to first record from BFR

                 int count=rs.getInt(1); // get first col value of that first record

                 //process the result
                 if(count==0)
                     System.out.println("INVALID CREDENTIALS");
                 else
                     System.out.println("VALID CREDENTIALS");
             }//if
            }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            //close jdbc obj
            try {
                if(rs!=null)
                    rs.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }
            try {
                if(ps!=null)
                    ps.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }
            try {
                if(con!=null)
                    con.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }

            try {
                if(sc!=null)
                    sc.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }//finally

    }//main
}//class
```

note:: make sure that "C:\oracleexeapp\oracle\product\11.2.0\server\bin" is added to PATH env. variable

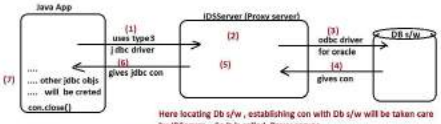
note:: Eclipse may give problems... try outside the eclipse

Type3 jdbc driver with Proxy server support
==========

Proxy server :: the server that interact with DB s/w using ODBC driver and establish the connection is called on behalfof Client app is called Proxy server.

eg:: IDS server

IDS server
==========
type ::proxy server
vendor : MicroSoft
default port no: 12
Gives type3 JDBC driver as built--in JDBC driver
To dowload s/w :: http://www.idssoftware.com/download.html



Here locating Db s/w , establishing con with Db s/w will be taken care by IDServer .. So it is called Proxy server.

=>WebServer can use type1/2/4/5 driver to locate and interact with DB s/w and also to establish the connection
=>Proxy can use ODBC driver to locate and interact with DB s/w and also to establish the connection.

Type3 jdbc driver details given by IDS server
==========
driver class name :: ids.sql.IDSDriver
of IDSServer
url :: jdbc:ids://<host>:<port>/conn?dsn=<system dsn>
protocol of ODBC Driver
location :: <IDSServer_HOME>\classes folder

IDS server :: Internet DB Access server

=>Every ODBC driver s/w is identified with its DSN (data source name)
there are 3 types of DSNS
a) user DSN (specific to currently logged in windows user)
b) System DSN (common for all the windows users of a computer)
c) File DSN (sharable in network)

note:: DSN should created by following process

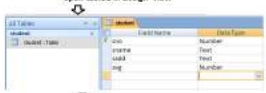
=>MsAccess DB s/w is part Ms-office s/w.

Procedure to create Logical DB having db table with records in Ms-Access
==========

Launch Ms Access --> menu Button -->new

(No username,password is required to connect to Ms-Access DB s/w becoz it is single user DB s /w)

open table1 in design view



ctrl+w

(open student db table to enter the records)

Every ODBC driver is identified and activated using DSN ..

PRocedure to create System DSN to odbc driver for Ms-Access
==========

control panel --> system and security --> admin tools --> ODBC Datasources (32 bit) --> system dsn --> select MS-Access Driver (*.accdb) -->

Exemple App Using Type3 Driver in support with IDSServer
==========

step1) keep IDS Server running
step2) keep MS- access dB s/w having db table with records (college.accdb with studen db table)
step3) keep System DSN ready for MsAccess Access driver (accdsn)
step4) add <IDSServer_home>\classes folder to eclipse build path..

right click on project -->build path --> libraries tab--> add class external class folder -->select <IDSServer_home>\classes

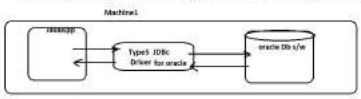
step5) gather type3 jdbc driver details

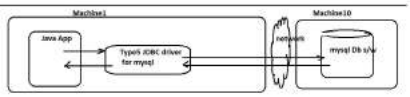
step6) Develop the the App

```
package com.nt.jdbc1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Type3DriverSelectTest {
    private static final String SELECT_STUDENTS_QUERY="SELECT SNO,SNAME,SADD,AVG FROM STUDENT";
        public static void main(String[] args) {
            try {
                //Load JDBc driver class (optional)
                Class.forName("ids.sql.IDSDriver");
            }
            catch(ClassNotFoundException cnf) {
                cnf.printStackTrace();
            }
            try(Connection con=DriverManager.getConnection("jdbc:ids://localhost:12/conn?dsn=accdsn");
                    Statement st=con.createStatement();
                    ResultSet rs=st.executeQuery(SELECT_STUDENTS_QUERY);
                    ){
                  //process the ResultSet object
                  if(rs!=null) {
                     while(rs.next()) {
                         System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                     }//while
                  }//if
                  }//try
              catch(SQLException se) {
                  se.printStackTrace();
              }
              catch(Exception e) {
                  e.printStackTrace();
              }
        }//main
}//class
```

Type5 JDBC driver
=================
=>It is not a official architecture for developing JDBC driver s/w..
=>It is given by Third party vendors and not approved by sun Ms
=> its architecture is very much similar to Type4 JDBC driver architecture

Advantages of Type5 JDBC driver
=============================
1) All in One Deployment :: Some Type4 JDBC drivers are coming as multiple jar files where
all the Type5 JDBC driver s/ws are coming as single jar file
2) Better performance :: Type5 drivers gives consistent performence when compare to
Type4 JDBC drivers [yet to be tested]
3) Codeless enhancement :: Type5 jdbc drivers s/w can be enhanced as required for ORM t/ws
by using some xml files based cfgs....
[All these advantages are claimed by Vendors of Type5 driver.. but not officially approved by Sun Ms]





=>The company "DataDirect" is providing Type5 JDBC driver s/ws for different Db s/ws.. as commercial
JDBC drivers having free trail period

=>DataDirect Type5 jdbc driver for oracle
=>DataDirect Type5 jdbc driver for mysql
=>DataDirect Type5 jdbc driver for postgreSQL
and etc..

DataDirect Type5 jdbc driver for oracle
Target DB s/w :: oracle
driver class name :: com.ddtek.jdbc.oracle.OracleDriver
url :: jdbc:datadirect:oracle://<host>:<port>;serviceName=<sid>
jar file :: oracle.jar (Big process download to jar files)
(supports auto loading jdbc driver class)

Example app
==========
step1) add oracle.jar [DataDirect's type5 jdbc driver for oracle) to the buildpath of eclipse project
(collect from our GIT HUB)
step3) Develop the Application as shown below..

```
public class Type5DriverSelectTest {
 private static final String SELECT_STUDENTS_QUERY="SELECT SNO,SNAME,SADD,AVG FROM STUDENT";
    public static void main(String[] args) {
        /*    try {
                    //Load JDBc driver class (optional)
                    Class.forName("com.ddtek.jdbc.oracle.OracleDriver");
                }
                catch(ClassNotFoundException cnf) {
                    cnf.printStackTrace();
                }*/
        try(Connection con=DriverManager.getConnection("jdbc:datadirect:oracle://localhost:1521;serviceName=xe",
                                                       "system","manager");
                    Statement st=con.createStatement();
                    ResultSet rset=st.executeQuery(SELECT_STUDENTS_QUERY);
                    ){
                //process the ResultSet object
                while(rs.next()) {
                    System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3)+"  "+rs.getFloat(4));
                }//while
                }//try
            catch(SQLException se) {
                se.printStackTrace();
            }
            catch(Exception e) {
                e.printStackTrace();
            }
    }//main
}//class
```

MetaData
========
=>Data about data or info about info
=>It can be either providing more data about given or gathering more data about given data.
=> The JDBC MetaData programming is given to gather more info about underlying DB tables and
Db s/w i.e we can get limitations and capabilities of underlying Db tables and DB s/ws.

3 types of JDBC Metdata Programmings
=> DatabaseMetaData
=> ResultsetMEtaData
=> ParameterMetaData (Most of jdbc drivers are not supporting this)

DatabaseMEtaData
================
=>Useful to get limitations and capabilities of underlying DB s/w.
=>DatabaseMetaData object means.. It is the object of underlying JDBc driver s/w supplied java
class implementing [java.sql.DatabaseMetaData]).
=>To create DatabaseMetaData object
DatabaseMetaData dbmd=con.getMetaData();
we can invoke multiple methods on dbmd to get various details about underlying Db s/w..
and that info changes DB s/w to Db s/w.

usecase :: useful while creating GUI DB tables like Toad for oracle , SQL developer and etc..

Example App
==========

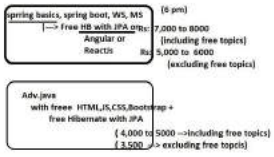
```
public class DatabaseMetaDataDemo {

    public static void main(String[] args) {

        try(//Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
                Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root", "root");
                ){
            //create DatabaseMetaData obj
            DatabaseMetaData dbmd=con.getMetaData();
            if(dbmd!=null) {
                System.out.println("db s/w name ::"+dbmd.getDatabaseProductName());
                System.out.println("DB s/w version::"+dbmd.getDatabaseProductVersion());
                System.out.println("jdbc driver name::"+dbmd.getDriverName());
                System.out.println("jdbc driver version::"+dbmd.getDriverVersion());
                System.out.println("All SQL keywords ::"+dbmd.getSQLKeywords());
                System.out.println("ALL numeric functions::"+dbmd.getNumericFunctions());
                System.out.println("All system functions ::"+dbmd.getSystemFunctions());
                System.out.println("All String functions ::"+dbmd.getStringFunctions());
                System.out.println(" Max chars in table name :: "+dbmd.getMaxTableNameLength());
                System.out.println("Max tables in a select Query:: "+dbmd.getMaxTablesInSelect());
                System.out.println(" Max row size:: "+dbmd.getMaxRowSize());
                System.out.println(" Supports PL/SQL procedures ?:: "+dbmd.supportsStoredProcedures());
            }
        }
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }

}
```

output
======

```
db s/w name ::MySQL
DB s/w version:: 8.0.22
jdbc driver name::MySQL Connector/J
jdbc driver version::mysql-connector-java-8.0.25 (Revision: 08be9e9b4cba6aa115f9b27b215887af40b159e0)
All SQL keywords
::ACCESSIBLE,ADD,ANALYZE,ASC,BEFORE,CASCADE,CHANGE,CONTINUE,DATABASE,DATABASES,DAY_HOUR,DAY_MICROSECOND,DAY_MINUTE,DAY_SECOND,DELAYED,DESC,DISTINCTROW,DIV,DUAL,ELSEIF,EMPTY,ENCLOSED,ESCAPED,EXIT,EXPLAIN,FIRST_VALUE,FLOAT4,FLOAT8,FORCE,FULLTEXT,GENERATED,GROUPS,HIGH_PRIORITY,HOUR_MICROSECOND,HOUR_MINUTE,HOUR_SECOND,IF,IGNORE,INDEX,INFILE,INT1,INT2,INT3,INT4,INT8,IO_AFTER_GTIDS,IO_BEFORE_GTIDS,ITERATE,JSON_TABLE,KEY,KEYS,KILL,LAG,LAST_VALUE,LEAD,LEAVE,LIMIT,LINEAR,LINES,LOAD,LOCK,LONG,LONGBLOB,LONGTEXT,LOOP,LOW_PRIORITY,MASTER_BIND,MASTER_SSL_VERIFY_SERVER_CERT,MAXVALUE,MEDIUMBLOB,MEDIUMINT,MEDIUMTEXT,MIDDLEINT,MINUTE_MICROSECOND,MINUTE_SECOND,MOD_WRITE_TO_BINLOG,NTH_VALUE,NTILE,OPTIMIZE,OPTIMIZER_COSTS,OPTION,OPTIONALLY,OUTFILE,PURGE,READ,READ_WRITE,REGEXP,RENAME,REPEAT,REPLACE,REQUIRE,RESIGNAL,RESTRICT,RLIKE,SCHEMA,SCHEMAS,SECOND_MICROSECOND,SEPARATOR,SHOW,SIGNAL,SPATIAL,SQL_BIG_RESULT,SQL_CALC_FOUND_ROWS,SQL_SMALL_RESULT,SSL,STARTING,STORED,STRAIGHT_JOIN,TERMINATED,TINYBLOB,TINYINT,TINYTEXT,UNDO,UNLOCK,UNSIGNED,USAGE,USE,UTC_DATE,UTC_TIME,UTC_TIMESTAMP,VARBINARY,VARCHARACTER,VIRTUAL,WHILE,WRITE,XOR,YEAR_MONTH,ZEROFILL
All numeric functions::ABS,ACOS,ASIN,ATAN,ATAN2,BIT_COUNT,CEILING,COS,COT,DEGREES,EXP,FLOOR,LOG,LOG10,MAX,MIN,MOD,PI,POW,POWER,RADIANS,RAND,ROUND,SIN,SQRT,TAN,TRUNCATE
All system functions ::DATABASE,USER,SYSTEM_USER,SESSION_USER,PASSWORD,ENCRYPT,LAST_INSERT_ID,VERSION
All String functions
::ASCII,BIN,BIT_LENGTH,CHAR,CHARACTER_LENGTH,CHAR_LENGTH,CONCAT,CONCAT_WS,CONV,ELT,EXPORT_SET,FIELD,FIND_IN_SET,HEX,INSERT,INSTR,LCASE,LEFT,LENGTH,LOAD_FILE,LOCATE,LOCATE,LOWER,LPAD,LTRIM,MAKE_SET,MATCH,MID,OCT,OCTET_LENGTH,ORD,POSITION,QUOTE,REPEAT,REPLACE,REVERSE,RIGHT,RPAD,RTRIM,SOUNDEX,SPACE,STRCMP,SUBSTRING,SUBSTRING,SUBSTRING,SUBSTRING,SUBSTRING,SUBSTRING_INDEX,TRIM,UCASE,UPPER
Max chars in table name :: 64
Max tables in a select Query :: 256
Max row size:: 2147483639
Supports PL/SQL procedures ?:: true
```

3 types of MetaDatas in JDBC
==========================
a)DatabaseMetaData
b) ResultSetMetaData
c) ParameterMetaData

ResultSetMetaData
==========
=> This more info about db table represented by ResultSet obj..
=> Using this ResultSetMetaData, we get col can names, cols count, col types and other details about db table..
=>ResultSetMetaData obj means it is the object of underlying JDBC driver s/w supplied java class that implements java.sql.ResultSetMetaData(I).
=>To create this object
ResultSet rs=st.executeQuery("SELECT * FROM STUDENT");
ResultSetMetaData rsmd=rs.getMetaData();



Modifier and Type	Method and Description
String	getCatalogName(int column) Gets the designated column's table's catalog name.
String	getColumnClassName(int column) Returns the fully-qualified name of the Java class whose instances are manufactured if this ResultSet object's getObject method 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	getColumnName(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.
String	getSchemaName(int column) Get the designated column's table's schema.
String	getTableName(int column) Gets the designated column's table name.
boolean	isAutoIncrement(int column) Indicates whether the designated column is automatically numbered.
boolean	isCaseSensitive(int column) Indicates whether a column's case matters.
boolean	isCurrency(int column) Indicates whether the designated column is a cash value.
boolean	isDefinitelyWritable(int column) Indicates whether a write or insert on the designated column will definitely succeed.

usecases :: => For report generation (To display table records with col names and data types)
=> In the development of GUI Db tables..

```
public class ResultSetMetaDataTest {

    public static void main(String[] args) {

        try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
                Statement st=con.createStatement();
                ResultSet rs=st.executeQuery("SELECT * FROM STUDENT");
                ){
            //create ResultSetMetaData obj
            ResultSetMetaData rsmd=null;
            if(rs!=null)
                rsmd=rs.getMetaData();
            //get column count
            if(rsmd!=null) {
                int colCount=rsmd.getColumnCount();
                for(int i=1;i<=colCount;++i) {
                    System.out.print(rsmd.getColumnName(i)+"   ");
                }
                System.out.println();
                for(int i=1;i<=colCount;++i) {
                    System.out.print(rsmd.getColumnTypeName(i)+"   ");
                }
                System.out.println();
                while(rs.next()) {
                    for(int i=1;i<=colCount;++i) {
                        System.out.print(rs.getString(i)+"   ");
                    }  //for
                    System.out.println();
                }//while

            }//if
        }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }//main
}//class
```

To print col names
To print col data types
To print col values

ParameterMetaData
==========
=> Gives more info about the parameters(?) that are there in pre-compiled SQL queries represented by the PreparedStatement, CallableStatement objs.
=>ParameterMetaData object means it is the obect underlying JDBC driver s/w supplied java class that implements java.sql.ParameterMetaData(I).
=> To create this object
PreparedStatement ps=con.prepareStatement("INSERT INTO STUDENT VALUES(?,?,?,?)");
ParmaeterMetaData pmd=ps.getParameterMetaData();

Method Summary

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 object could supply 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.

=>Most of popular JDBC drivers are not supporting ParameterMetaData Programming becoz they have not implemented ParameterMetaData(I) and its methods properly (U may get UnSupportedFeatureException)

```
public class ParameterMetaDataTest {

    public static void main(String[] args) {

        try(Connection con=DriverMannager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system", "manager");
                PreparedStatement ps=con.prepareStatement("INSERT INTO STUDENT VALUES(?,?,?,?)");
                ){
            //create ParameterMetaData  obj
            ParameterMetaData pmd=ps.getParameterMetaData();
            if(pmd!=null) {
                int paramsCount=pmd.getParameterCount();
                for(int i=1;i<=paramsCount;++i) {
                    System.out.println("parameter number::"+i);
                    System.out.println("parameter name mode::"+pmd.getParameterMode(i));
                    System.out.println("parameter type name::"+pmd.getParameterTypeName(i));
                    System.out.println("parameter is singned::"+pmd.isSigned(i));
                }//for
            }//if
        }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }//main
}//class
```

This app throws
java.sql.SQLException: Unsupported feature
becoz oracle thin driver is not implementing this feature

note:: In Realtime , we ResultSetMetaData concept a alot in the report generation to display col names and values

Procedure use GUI Builder plugin to develope AWT/SWING Apps throuhg drag and drop operations

Plugin :: It is patch s/w that provides additional functionalaties to the existing software.

=>Eclipse IDE gives basic features.. and More features can be added through plugins
=> Eclipse supplied plugins can be added Eclipse IDE using " Install new software" option of help menu.
=> Third party s supplied plugins can be added Eclipse IDE using " Market place" option of help menu.

step1) Install GUI Builder plugins and restart IDE..

Help menu --->Install new s/w ---> select url (2021-03 - http://download.eclipse.org/releases/2021-03) -->select general purpose tools --->
select the following items

-->next --> accept terms and conditions ---> ask for restarting IDE --> restart IDE.

step2) Make sure that Project is having odbc6/7/8.jar file

step3) Create Frame window app
right click on packge --->new ---> other ---> window builder--> swing designer --->application window --->
name :: GUIBuilder_ScrollFrame

step4) Use design mode and Design the app through drag and drop
a) set Layout of the App
Right click on window --->set layout --> absolute layout..

step5) add ActionEvent related ActionListener for all the 4 buttons

right click on button --->add event handler --->action --> action performed (do it for all the 4 buttons)

step6) Develop intializeJDBC() method as private method having logic to create JDBC objs including Scrollable RS and call method from constructor.

In the source tab of the Application

```
private Connection con;
    private PreparedStatement ps;
    private ResultSet rs;
public GUIBuilder_ScrollFrame() {
        initialize();
        initializeJDBC();
    }

    private void initializeJDBC() {
        try {
            con=DriverManager.getConnection("jdbc:oracle:thin@localhost:1521:xe", "system", "manager");
            ps=con.prepareStatement(SELECT_STUDENT_QUERY,
                                    ResultSet.TYPE_SCROLL_SENSITIVE,
                                    ResultSet.CONCUR_READ_ONLY);
            rs=ps.executeQuery();
        }
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }//initializeJDBC()
```

step7) write following code in the actionPerformed(-) method of first button

```
JButton btnNewButton = new JButton("first");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    rs.first();
                    textField.setText(rs.getString(1));
                    textField_1.setText(rs.getString(2));
                    textField_2.setText(rs.getString(3));
                    textField_3.setText(rs.getString(4));
                }
                catch(SQLException se) {
                    se.printStackTrace();
                }
            }
        });
```

22nd : adv.java (4pm)
two entries
1) from JDBC (prerequisite:: Core java oops)
2) from servlet (prerequisite:: Core java oops)

now onwards
for adv.java ==> hibernate course is free +html ,js, css free
forspring basics, spring boot and MS => angular/react /hibernate free
(August 12th 6pm)

step7) write following code in the actionPerformed(-) method of next button

```
JButton btnNewButton_1 = new JButton("next");
        btnNewButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    if(!rs.isLast()) {
                        rs.next();
                        textField.setText(rs.getString(1));
                        textField_1.setText(rs.getString(2));
                        textField_2.setText(rs.getString(3));
                        textField_3.setText(rs.getString(4));
                    }//if
                }//try
                catch(SQLException se) {
                    se.printStackTrace();
                }
            }
        });
```

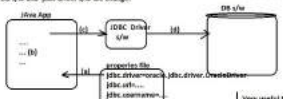
assignment :: write for remaining 2 Buttons..

step8) add WindowListner event handler

right click title bar of Application window ---> window ---> windowClosing--->
go source tab -->write the following in windowClosing(-) method..

```
frame.addWindowListener(new WindowAdapter() {
            @Override
            public void windowClosing(WindowEvent e) {
                System.out.println("GUIBuilder_ScrollFrame.initialize().new WindowAdapter() {...}.windowClosing()");
                try {
                    if(rs!=null)
                        rs.close();
                }
                catch(SQLException se) {
                    se.printStackTrace();
                }
                try {
                    if(ps!=null)
                        ps.close();
                }
                catch(SQLException se) {
                    se.printStackTrace();
                }
                try {
                    if(con!=null)
                        con.close();
                }
                catch(SQLException se) {
                    se.printStackTrace();
                }


            }
        });
```

Properties file
=============
=>The text file that maintains entries in the form of key=value pairs is called properties file
=> The standard ways of software industry is do not hard code input values in the App... always pass them to the App from outside the app through properties file.
=> The driver class name , url , db username,password are called JDBC properties.. It is recomended to pass them app from outside the App by taking the support of properties file. So we can change jdbc properties based on the DB s/w and jdbc driver s/w we change.

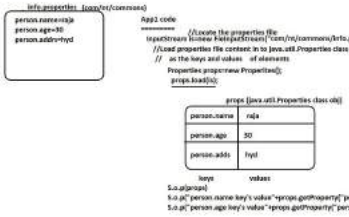


[This properties file gives flexibility of modification towards the change of DB s/w and JDBC driver s/w]

Very useful towards release of the Project to change jdbc properties details according to the Db s/w we use.

---

java.util.Properties
==================
-util is map collection and the sub class of java.util.Hashtable
=>The speciality of this map collection allows only String as keys ,values of elements
(This is default restriction with out using generics)
=> The keys and values of elements can be gathered directly from text properties using props.load(..) method.



If we forget password certain oracle DB user to reset to new password
===

```
SQL> conn as sysdba
Enter user-name: system
Enter password: iyh
Connected.
SQL> alter user system identified by manager;
                                     any new password

User altered.

SQL> conn system/manager
Connected.
SQL>
```

If unfortunately Oracle user Accout is locked then use
SQL> Alter user "<userName>" identified by "NewPass" Account Unlock;

---

jdbc.properties (com/nt/commons)
==========================

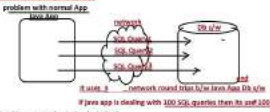
```
#contains jdbc properties
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@localhost:1521:xe
jdbc.username=system
jdbc.pwd=manager
```

ScrollableRSTestWithPropertiesFile.java

```
package com.nt.jdbc;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

public class ScrollableRSTestWithPropertiesFile {
    private static final String EMP_SELECT_QUERY="SELECT EMPNO,ENAME,JOB,SAL FROM EMP";
    public static void main(String[] args) {
        Properties props=null;
        try(InputStream is=new FileInputStream("src/com/nt/commons/jdbc.properties")){
            props=new Properties();
            props.load(is);
        }
        catch(FileNotFoundException fnf) {
            fnf.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

        try(Connection con=DriverManager.getConnection(props.getProperty("jdbc.url"),
                                                       props.getProperty("jdbc.username"),
                                                       props.getProperty("jdbc.pwd"));
            Statement st=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
            ResultSet rs=st.executeQuery(EMP_SELECT_QUERY);
            ){
            if(rs!=null) {
                System.out.println("RS records top to bottom ");
                while(rs.next()) {
                    System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                }
                System.out.println("-------------------------");
                System.out.println("RS records bottom to top ");
                rs.afterLast();
                while(rs.previous()) {
                    System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                }
                rs.first();
                System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");
                rs.last();
                System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");
                rs.absolute(2);
                System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");
                rs.absolute(-4);
                System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");
                rs.relative(2);
                System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");
                rs.relative(-3);
                    System.out.println(rs.getRow()+"----->"+rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3)+" "+rs.getFloat(4));
                System.out.println("-------------------------");

                }//if

        }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }

}
```

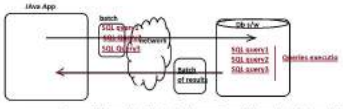
Batch Processing/Batch Updation

problem with normal App



If java app is dealing with 100 SQL queries then its and 100 network roundtrips b/w java app and Db s/w.

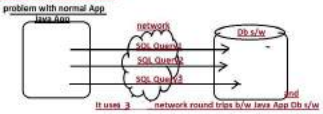
Batching process/updation (solution)
==========
Combines multiple SQL queries into single batch and sends that batch to Db s/w at once , executes the SQL queries in Db s/w one by one from batch and gets SQL query results back to Java App as batch



we can add any no.of SQL queries to the batch.. since sending multiple queries to Db s/w as batch and recieving results as batch we can say batch processing reduces network round trips b/w Java App and Db s/w. (only one network round trip will be used)

Batch Processing/BatchUpdation

problem with normal App

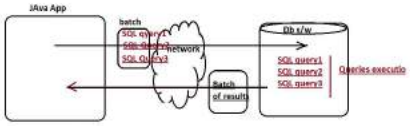


It uses 3 network round trips b/w Java App Db s/w

If java app is dealing with 100 SQL queries then its use 100 network rountrips b/w java app and Db s/w.

Batching process/updation (solution)

Combines multiple SQL queries into single batch and sends that batch to Db s/w at once , executes the SQL queries in Db s/w one by one from batch and gets SQL query results back to Java App as batch



we can add any no.of SQL queries to the batch.. since sending multiple queries to Db s/w as batch and recieving results as batch we can say batch processing reduces network round trips b/w java App and Db s/w. (only one network round trip will be used)

=>We can add only multiple non-select SQL queries to the batch using addBatch(-) method.. and use executeBatch() method to send batch of queries to dB s/w at once.. to execute them in DB s/w and get results as batch back to JAva App.

=>Non-Select SQL queries are also called as update query becoz they do one or other updation to db table ..,Batch Procesing can be called Batch updation technically.

Process:

step1) add multiple non-selct queries to the batch

```
//add queries to batch ..these queries can belong to same db table or different db tables but must be non-select Queries
st.addBatch("INSERT INTO STUDENT VALUES(6189,'animesh','delhi',78.99)");
st.addBatch("UPDATE STUDENT SET AVG=AVG-40 WHERE SNO>=10000");
st.addBatch("DELETE FROM STUDENT WHERE SNO<=10");
```

any no.of and any type of non-select Queries belong to either same db table or different db tables can be added to the batch of Batch processing.

step2) send and execute the batch of SQL queries in Db s/w

```
//execute the batch
int result[]=st.executeBatch();
```

step3) process the results

```
int sum=0;
    for(int i=0;i<result.length;++i)
        sum=sum+result[i];
System.out.println("Total no.of records that effected::"+sum);
```

note: By default batch processing queries do not execute by appying atomocity i.e if one or another query of batch fails to execute it does not effect other queries execution. To bring the effect , we need to use Transaction Management.

Why we can not add "SELECT SQL Queries" to the batch?

Ans) The select SQL query generates JDBC ResultSet obj and the return type st.executeBatch() method is int[] i.e we can not place ResultSet obj ine int[] ..Due to this we can say select SQL queries can not be added to the batch of batch processing

Can u explain different execute() methods that can be invoked using statement obj?

a) ResultSet executeQuery(-) :: To send and execute SELECT SQL queries in db table

b) int executeUpdate(-) :: To send and execute NON-SELECT SQL queries in db tables which manipulates integer range records in db table

c) long executeLargeUpdate(-) :: To send and execute NON-SELECT SQL queries in db tables which manipulates long range records in db tables

d) boolean execute(-) :: To send and excute both SELECT, NON-SELECT Queries and also for calling PL/SQL procedures and functions of DB s/w

e) int[] executeBath() :: To send and execute batch of SQL queries in Db s/w.

Transaction Management

=> The process of combining related operations in to single unit and executing them by applying do every thing or nothing principle is called Tx Mgmt (TransactionManagement)

=>Batch processing does not perform Tx Mgmt by default.. we need to write additional logics on the top of Batch Processing result to bring the effect of TxMgmt

usecase :: Transfer money is combination two operations
a) withdraw amount from source account
b) Deposite amount inot dest account
[If completed ,both shoud be completed otherwise nothing should be completed.

For Tx Mgmt through Batch Processing

```
//disable AutoCommit on Db s/w
  con.setAutoCommit(false);  (Begin Tx)

 // add queries to that batch                    |
  st.addBatch("........");                       |
  st.addBatch("........");                       |  Execute Statements
 //execute batch                                 |  in Tx
  int result[]=st.executeBatch();                |

//commit or rollback the Tx (Tx Mgmt)
  boolean flag=true;
  for(int i=0;i<result.length;++i){
     if(result[i]==0){
        flag=false;
        break;
     }
  }
  if(flag==false){
   con.rollback();
   s.o.println("Tx rollbacked");
  }
  else{
   con.commit();
   s.o.println("Tx committed");
  }
```

result[]: 1 → 0, 0 → 1

3 Imp operations of Tx Mgmt
a) Begin Tx
b) Executes statements in Tx
c) commit or RollbackTx

Example App on Transfer Money

```
//JansferMoneyTxMgmtTest
package com.nt.jdbc4;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/*CREATE TABLE "SYSTEM"."JDBC_ACCOUNT"
(    "ACNO" NUMBER(10,0) NOT NULL ENABLE,
     "HOLDER" VARCHAR2(20 BYTE),
     "BALANCE" FLOAT(126),
     CONSTRAINT "JDBC_ACCOUNT_PK" PRIMARY KEY ("ACNO"));
*/
public class TransferMoneyTxMgmtTest {

    public static void main(String[] args) {

        long srcAcno=0,destAcno=0;
        double amount=0.0;
        //read inputs
        try(Scanner sc=new Scanner(System.in)){
              if(sc!=null) {
                    System.out.println("enter soruce account number:");
                    srcAcno=sc.nextLong();
                    System.out.println("Enter dest account number ::");
                    destAcno=sc.nextLong();
                    System.out.println("Enter amount to transfer:");
                    amount=sc.nextDouble();
              }//if
        }//try
        catch(Exception e) {
            e.printStackTrace();
        }

        try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
                Statement st=con.createStatement();
                ){
            //Begin Tx
            if(con!=null)
              con.setAutoCommit(false);

              if(st!=null) {
                    //add queries to the batch  (executing queries in the batch)
                    //for withdraw opeation
                    st.addBatch("update JDBC_ACCOUNT set balance=balance-"+amount+" where acno="+srcAcno);
                    //for deposite opeation
                    st.addBatch("update JDBC_ACCOUNT set balance=balance+"+amount+"where acno="+destAcno);
                    //execut the batch
                    int result[]=st.executeBatch();

                    //process the results from TxMgmt
                    boolean flag=true;
                    for(int i=0;i<result.length;++i) {
                          if(result[i]==0) {
                                flag=false;
                                break;
                          }
                    }
                    if(flag==true) {
                          con.commit();  //commit the Tx  (TCL)
                          System.out.println("Tx Commited , Money Transffered");
                    }
                    else {
                          con.rollback();  //rollback the Tx   (TCL)
                          System.out.println("Tx rolledback , Money  not Transffered");
                    }
              }//if
        }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }//main
```

Rowsets

=>Rowsets are extension of ResultSets solving the limitations of ResultSets..
=>All Rowsets are the objs of Java classes Implementing javax.servlet.RowSet(I) directly or indirectly and RowSet(I) is the sub interface of ResultSet(I)..

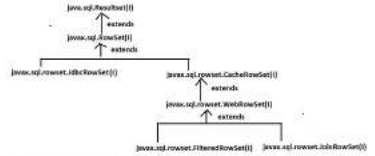
Limitations of ResultSets

(a) ResultSet objs are not serializable objects .. so they can not be sent over the network
(b) ResultSets are Connected objects i.e with out having connectivity with DB s/w we can not do any thing in the ResultSet object
(c) we can not perform Event handling on ResultSets
(d) Does not support bean style programming (setters methods, getters methods)
and etc..

=>We can send only Serializable objs over the network
=>Object becomes serializable only when the class of that object implements java.io.Serializable(I).
=>Serialization means converting object data into bits and bytes to send over the network or to write to files..

=>To overcome all these problems , take the support of RowSets

5 Types of Rowsets
(a) JdbcRowSets
(b) CachedRowSets
(c) WebRowsets
(d) JoinRowsets
(e) FilteredRowSets



Oracle thin driver gives support for all the 5 Types of Rowsets .. they are
a) OracleJdbcRowSet
b) OracleCachedRowSet
c) OracleWebRowSet
d) OracleJoinRowSet
e) OracleFiltredRowSet

Advantages RowSets

=>except JdbcRowSets , the remaining all rowsets are Serializable Rowsets i.e their objects data can be sent over the network.
=>except JdbcRowSets , the remaining all rowsets DisConnected Rowsets i.e their data can be modified with out having connection with DB s/w. (This indicates offline processing is possible with RowSets)
=>supports Bean style programming
=> Supports event handling...

JDBC RowSets

=>These types of Rowsets are ResultSets in RowSet env..
=>Supports bean style programming
=>These are connected objects (With out having connection with Db s/w, we can not manipulate their data)
=>These are not serializable objects..

```
//JdbcRowSetDemo.java
package com.nt.jdbc4;
import java.sql.SQLException;
import oracle.jdbc.rowset.OracleJDBCRowSet;
public class JdbcRowSetDemo {
      public static void main(String[] args) {
            try(OracleJDBCRowSet jrowset=new OracleJDBCRowSet()){
Bean style  jrowset.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
programing  jrowset.setUsername("system");
            jrowset.setPassword("manager");
            jrowset.setCommand("SELECT * FROM STUDENT");
            jrowset.execute();
            while(jrowset.next()) {
                  System.out.println(jrowset.getInt(1)+" "+jrowset.getString(2)+" "+jrowset.getString(3)+" "+jrowset.getString(4));
            }//while
            }
            catch(SQLException se) {
                  se.printStackTrace();
            }
            catch(Exception e) {
                  e.printStackTrace();
            }

      }//main
}//class
```

Transaction Management

=> The process of combining related operations in to single unit and executing them by applying do every thing or nothing principle is called Tx Mgmt (TransactionManagement)

=>Batch processing does not perform Tx Mgmt by default.. we need to write additional logics on the top of Batch Processing result to bring the effect of TxMgmt

usecase :: Transfer money is combination two operations
a) withdraw amount from source account
b) Deposite amount inot dest account
[If completed ,both shoud be completed otherwise nothing should be completed.

For Tx Mgmt through Batch Processing

```
//disable AutoCommit on Db s/w
  con.setAutoCommit(false);  (Begin Tx)

 // add queries to that batch                    |
  st.addBatch("........");                       |
  st.addBatch("........");                       |  Execute Statements
 //execute batch                                 |  in Tx
  int result[]=st.executeBatch();                |

//commit or rollback the Tx (Tx Mgmt)
  boolean flag=true;
  for(int i=0;i<result.length;++i){
     if(result[i]==0){
        flag=false;
        break;
     }
  }
  if(flag==false){
   con.rollback();
   s.o.println("Tx rollbacked");
  }
  else{
   con.commit();
   s.o.println("Tx committed");
  }
```

result[]: 1 → 0, 0 → 1

3 Imp operations of Tx Mgmt
a) Begin Tx
b) Executes statements in Tx
c) commit or RollbackTx

Example App on Transfer Money

```
//JansferMoneyTxMgmtTest
package com.nt.jdbc4;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

/*CREATE TABLE "SYSTEM"."JDBC_ACCOUNT"
(    "ACNO" NUMBER(10,0) NOT NULL ENABLE,
     "HOLDER" VARCHAR2(20 BYTE),
     "BALANCE" FLOAT(126),
     CONSTRAINT "JDBC_ACCOUNT_PK" PRIMARY KEY ("ACNO"));
*/
public class TransferMoneyTxMgmtTest {

    public static void main(String[] args) {

        long srcAcno=0,destAcno=0;
        double amount=0.0;
        //read inputs
        try(Scanner sc=new Scanner(System.in)){
              if(sc!=null) {
                    System.out.println("enter soruce account number:");
                    srcAcno=sc.nextLong();
                    System.out.println("Enter dest account number ::");
                    destAcno=sc.nextLong();
                    System.out.println("Enter amount to transfer:");
                    amount=sc.nextDouble();
              }//if
        }//try
        catch(Exception e) {
            e.printStackTrace();
        }

        try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
                Statement st=con.createStatement();
                ){
            //Begin Tx
            if(con!=null)
              con.setAutoCommit(false);

              if(st!=null) {
                    //add queries to the batch  (executing queries in the batch)
                    //for withdraw opeation
                    st.addBatch("update JDBC_ACCOUNT set balance=balance-"+amount+" where acno="+srcAcno);
                    //for deposite opeation
                    st.addBatch("update JDBC_ACCOUNT set balance=balance+"+amount+"where acno="+destAcno);
                    //execut the batch
                    int result[]=st.executeBatch();

                    //process the results from TxMgmt
                    boolean flag=true;
                    for(int i=0;i<result.length;++i) {
                          if(result[i]==0) {
                                flag=false;
                                break;
                          }
                    }
                    if(flag==true) {
                          con.commit();  //commit the Tx  (TCL)
                          System.out.println("Tx Commited , Money Transffered");
                    }
                    else {
                          con.rollback();  //rollback the Tx   (TCL)
                          System.out.println("Tx rolledback , Money  not Transffered");
                    }
              }//if
        }//try
        catch(SQLException se) {
            se.printStackTrace();
        }
        catch(Exception e) {
            e.printStackTrace();
        }

    }//main
```

Rowsets

=>Rowsets are extension of ResultSets solving the limitations of ResultSets..
=>All Rowsets are the objs of Java classes Implementing javax.servlet.RowSet(I) directly or indirectly and RowSet(I) is the sub interface of ResultSet(I)..

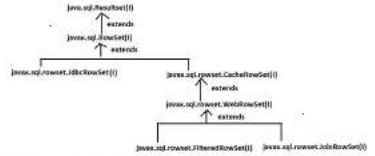
Limitations of ResultSets

(a) ResultSet objs are not serializable objects .. so they can not be sent over the network
(b) ResultSets are Connected objects i.e with out having connectivity with DB s/w we can not do any thing in the ResultSet object
(c) we can not perform Event handling on ResultSets
(d) Does not support bean style programming (setters methods, getters methods)
and etc..

=>We can send only Serializable objs over the network
=>Object becomes serializable only when the class of that object implements java.io.Serializable(I).
=>Serialization means converting object data into bits and bytes to send over the network or to write to files..

=>To overcome all these problems , take the support of RowSets

5 Types of Rowsets
(a) JdbcRowSets
(b) CachedRowSets
(c) WebRowsets
(d) JoinRowsets
(e) FilteredRowSets



Oracle thin driver gives support for all the 5 Types of Rowsets .. they are
a) OracleJdbcRowSet
b) OracleCachedRowSet
c) OracleWebRowSet
d) OracleJoinRowSet
e) OracleFiltredRowSet

Advantages RowSets

=>except JdbcRowSets , the remaining all rowsets are Serializable Rowsets i.e their objects data can be sent over the network.
=>except JdbcRowSets , the remaining all rowsets DisConnected Rowsets i.e their data can be modified with out having connection with DB s/w. (This indicates offline processing is possible with RowSets)
=>supports Bean style programming
=> Supports event handling...

JDBC RowSets

=>These types of Rowsets are ResultSets in RowSet env..
=>Supports bean style programming
=>These are connected objects (With out having connection with Db s/w, we can not manipulate their data)
=>These are not serializable objects..

```
//JdbcRowSetDemo.java
package com.nt.jdbc4;
import java.sql.SQLException;
import oracle.jdbc.rowset.OracleJDBCRowSet;
public class JdbcRowSetDemo {
      public static void main(String[] args) {
            try(OracleJDBCRowSet jrowset=new OracleJDBCRowSet()){
Bean style  jrowset.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
programing  jrowset.setUsername("system");
            jrowset.setPassword("manager");
            jrowset.setCommand("SELECT * FROM STUDENT");
            jrowset.execute();
            while(jrowset.next()) {
                  System.out.println(jrowset.getInt(1)+" "+jrowset.getString(2)+" "+jrowset.getString(3)+" "+jrowset.getString(4));
            }//while
            }
            catch(SQLException se) {
                  se.printStackTrace();
            }
            catch(Exception e) {
                  e.printStackTrace();
            }

      }//main
}//class
```

CachedRowset

[illegible]

WebRowset

[illegible]

JoinRowset

[illegible]

FilteredRowSet

[illegible]