dt : 27/4/2022(day-1)

part-1 : CoreJava

=>Alphabets(Programming Components)

(a)Variables

(b)Methods

(c)Blocks

(d)Constructors

(e)Classes

(f)Interfaces

(g)AbstractClasses

=>Programming Concepts

=>Object Oriented Programming features

Part-2 : AdvJava

define Aplication?

=>The set-of-programs collected together to perform define action is known as Application.

define Web Application?

=>The application which is executing in Web environment or Internet environment is known as Web Application.

=>We use the following technologies to construct WebApplications:

1.JDBC

2.Servlet

3.JSP

1.JDBC:

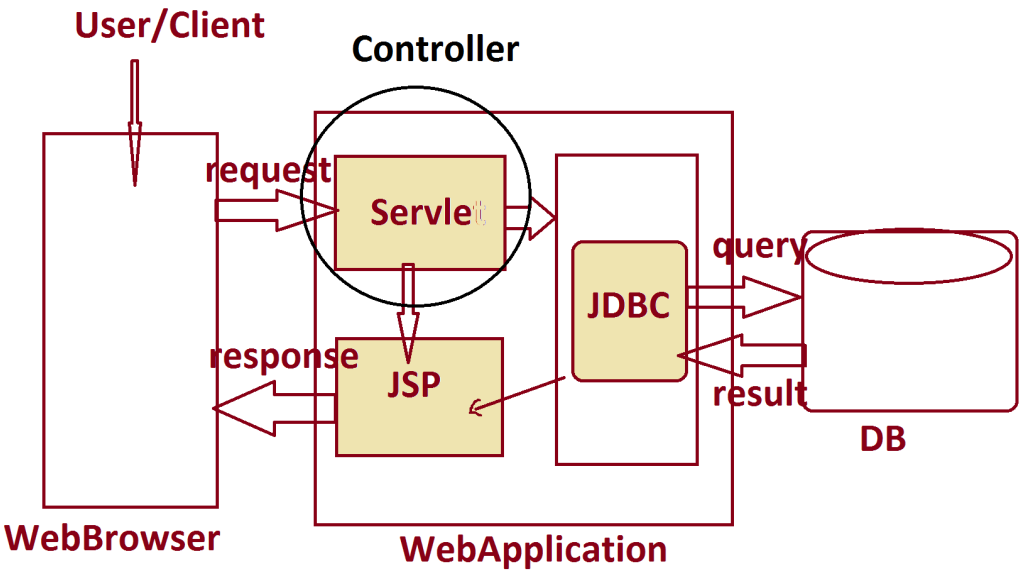
=>JDBC stands for 'Java DataBase Connectivity' and which is used to establish connection b/w JavaProgram and DB product.

2.Servlet:

=>Servlet means 'Server program' and which accept the request from User/Client and gives the response.

3.JSP:

=>JSP stands for 'Java Server Page' and which is response from WebApplication.



==========

1.JDBC:(Unit-1)

Types of Storages:

=>According to application development the Storages are categorized into four types:

(a)Field Storage

(b)Object Storage

(c)File Storage

(d)DataBase Storage

(a)Field Storage:

=>The memory generated to hold single data

value is known as Field Storage.

=>The primitive data types will generate

field Storage.

**imp*

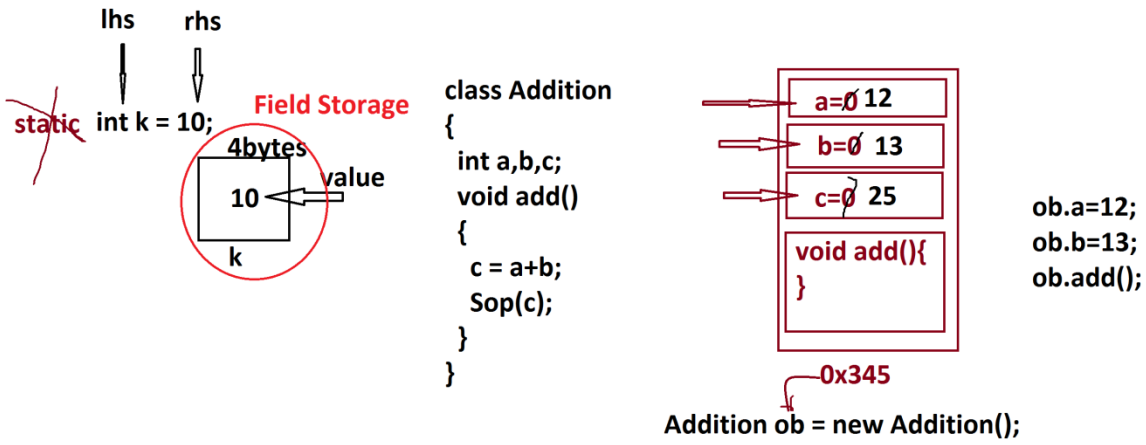
(b)Object Storage:

=>The memory generated to hold group members

is known as Object Storage.

```
class Addition
{
 int a,b,c;
 void add()
 {
  c = a+b;
  Sop(c);
 }
}

Addition ob = new Addition();
```



Dt : 28/4/2022(Day-2)

faq:

wt is the diff b/w

(i)Object

(ii)Object reference

(iii)Object reference Variable

(i)Object:

=>The memory generated to hold NonStatic members of class is known as Object.

(ii)Object reference:

=>The address location where the object is created is known as Object reference.

(iii)Object reference Variable:

=>The variable which is holding the object reference is known as 'Object reference Variable'.

Summary:

=>Variable in Java are categorized into two types:

(i)Primitive DataType variable

(ii)NonPrimitive DataType Variable

(Reference DataType Variable)

=>Primitive DataType variables will hold values and NonPrimitive DataType variables will hold Object references.

==========

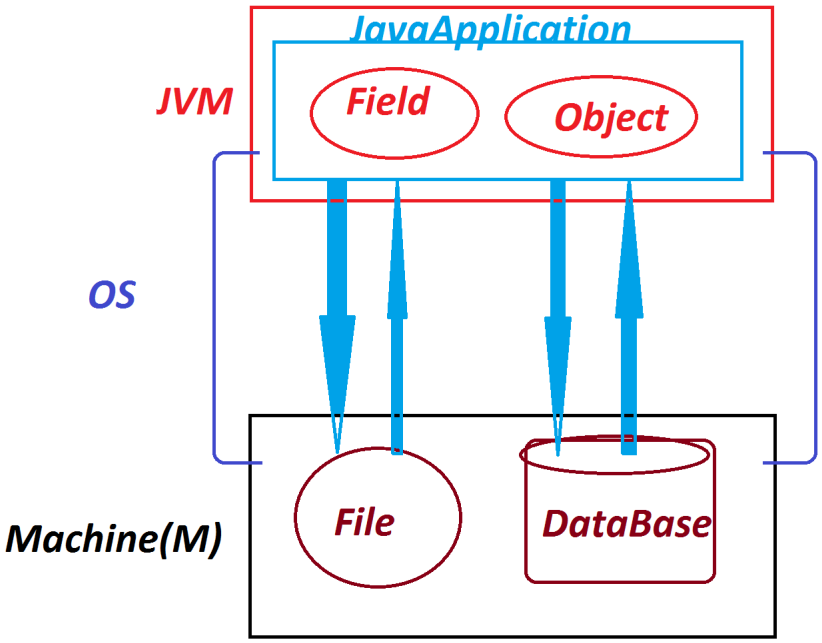
Note:

=>The field Storages and the Object Storages which are generated part of JVM while application execution will be destroyed automatically when the JVM ShutDowns.

=>when we want to have permanent storages for applications then we must take the support of any one of the following:

=>File Storage

=>DataBase Storage



==========

faq:

define API?

=>API stands for 'Application Programming Interface' and which is collection of related 'classes and Interfaces' used in constructing applications to interact with resources.

(According to Java API means package)

=>The following are some important packages available from JavaLib:

CoreJava:

(i)java.lang - Language package

(ii)java.io - Streams and Files package

(iii)java.util - Utility package

(iv)java.net - Networking package

AdvJava:

(i)java.sql - DataBase Connection package

(ii)javax.servlet - Servlet Programming

(iii)javax.servlet.jsp - JSP parogramming

==========

**imp*

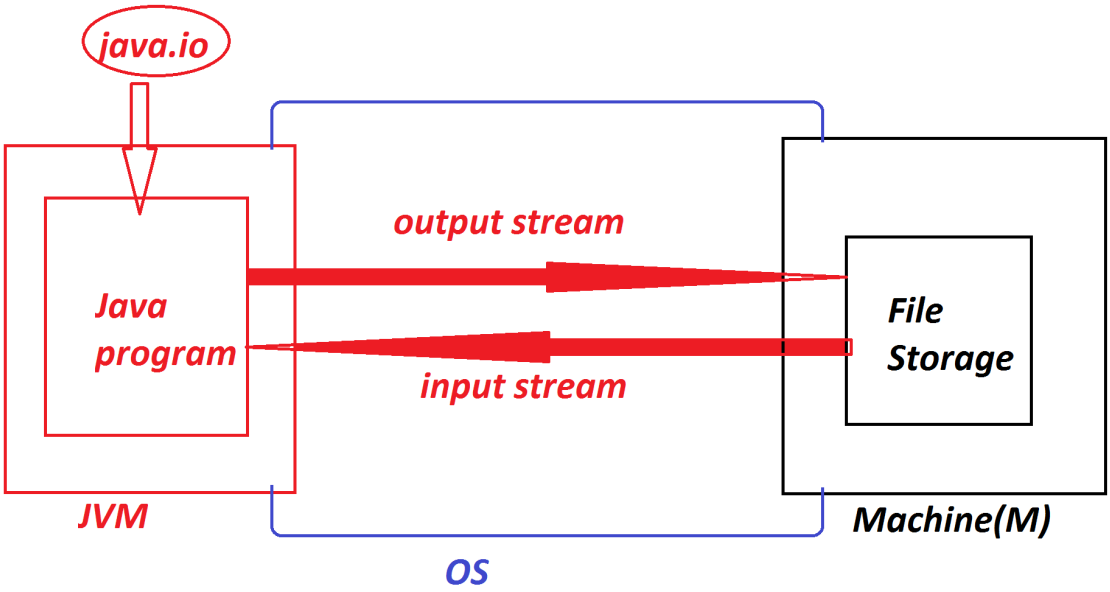
(c)File Storage:

=>The smallest permanent storage of computer System,which is controlled and managed by the OperatingSystem is known as File Storage.

Note:

=>In the process of establishing communication b/w JavaProgram and File Storage,the JavaProgram must be constructed using the classes and Interfaces available from java.io package.

Diagram:



faq:

define output stream?

=>The Java program writing the data to file storage is known as output stream.

faq:

define input stream?

=>The Java program reading the data from the file storage is known as input stream.

Dt : 29/4/2022

faq:

define Stream?

=>The contineous flow of data is known as Stream.

Types of Streams:

=>Streams in Java are categorized into two types:

1.Byte Stream

2.Character Stream

1.Byte Stream:

=>The contineous flow of data in the form of 8-bits is known as ByteStream or Binary Stream.

2.Character Stream:

=>The contineous flow of data in the form of 16-bits is known as Character Stream or text Stream.

Note:

=>Through Byte Stream we can send multimedia data files like Audio,Video,Image and Animation.

=>Through Binary Stream we can send text data or Character Stream.

==========

Dis-Advantages of File Storage:

1.Data Redundancy

2.Data Inconsistency

3.Difficuilty in Accessing Data

4.Limite data Sharing

5.Integrity Problems

6.Atomicity Problems

7.Concurrent Access Anomalies

8.Security problems

==========

Note:

=>Because of these Dis-Advantages we take the support of DataBase for JavaApplications.

==========

**imp*

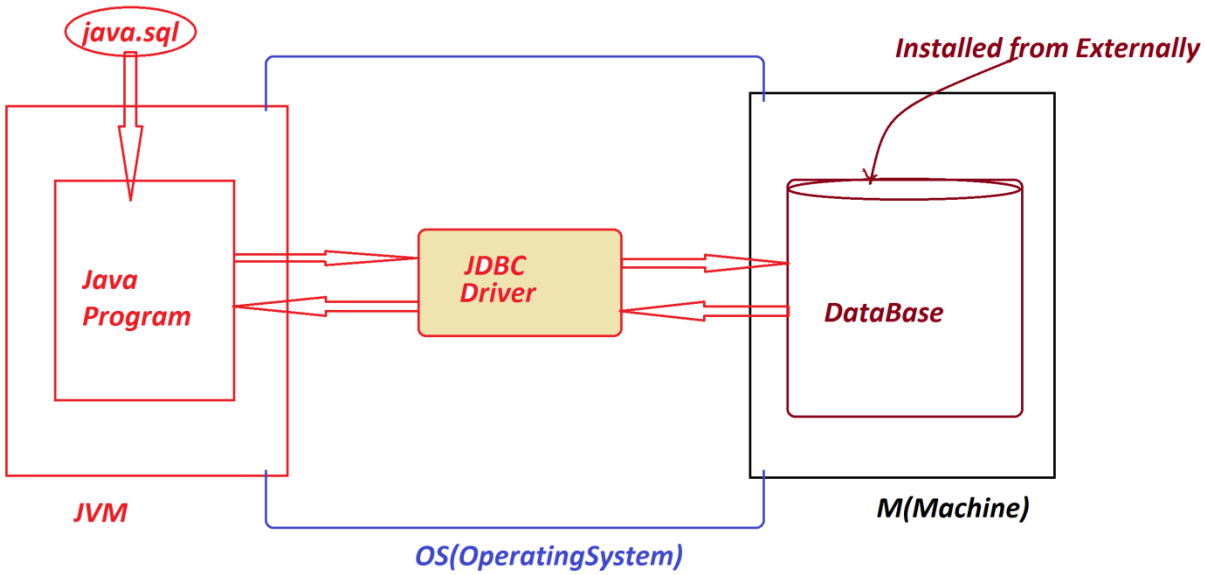
(d)DataBase Storage:

=>The largest Permanent Storage which is installed from externally into ComputerSystem is known as DataBase Storage.

Note:

=>In the process of establishing communication b/w JavaProgram and DataBase product,the JavaProgram must be constructed using 'classes and interfaces' available from 'java.sql' package and the Program must use 'JDBC driver'.

Diagram:



==========

define "driver"?

=>The small s/w program part of Operatingsystem which is used to control the resources of ComputerSystem is known as 'driver'.

Ex:

Audio driver

Video driver

N/W driver

...

define JDBC driver?

=>The driver which is used to communicate with DataBase product is known as JDBC driver(Java DataBase Connectivity Driver).

Types of JDBC drivers:

=>JDBC drivers are categorized into four types:

1.JDBC-ODBC bridge driver(Type-1)

2.Native API driver(Type-2)

3.Network Protocol driver(Type-3)

4.Thin driver(Type-4)

Note:

=>In realtime we use 'Thin driver' for application development.

==========

Dt : 30/4/2022

**imp*

Making System Environment ready to construct JDBC Application:

step-1 : Download and Install Oracle DataBase

step-2 : Check the Login process with Oracle DataBase

step-3 : Create table with name Product45

```
create table Product45(pcode varchar2(10),pname varchar2(15),
pprice number(10,2),pqty number(10),primary key(pcode));
```

step-4 : Insert records to DBTable Product45.

insert into Product45 values('A111','Mouse',1200,12);

*SQL> select * from Product45;*

PCODE PNAME PPRICE PQTY

---------- ---------- ---------- ----------

A111 Mouse 1200 12

A222 KBB 1100 10

A105 CDR 1300 13

A104 FDD 700 3

A100 CDR 1200 45

SQL>

step-5 : Find DB-Jar file and copy into one User defined folder.

DB Jar file is availabel at 'lib' folder of Oracle

C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib

**imp*

define DataBase JAR file?

=>JAR stands for 'Java Archieve' and which is compressed format

of class files.

Oracle - ojdbc14.jar,ojdbc6.jar, ojdbc7.jar, ojdbc8.jar,

ojdbc10.jar

Oracle10 - ojdbc14.jar

Oracle11 - ojdbc6.jar

oracle12 - ojdbc7.jar,ojdbc8.jar

other - ojdbc10.jar

MySQL - mysql-connector-java-VERSION.jar

SQL Server - sqljdbc41.jar, sqljdbc42.jar

PostgreSQL - postgresql-VERSION.jar

Apache Derby - derby.jar, derbyclient.jar

SQLite - sqlite-jdbc-VERSION.jar

Microsoft Access - ucanaccess-VERSION.jar

step-6 : Find Oracle DataBase PortNo and ServiceName

=>Open 'tnsnames.ora' file from "ADMIN" folder of "NETWORK" to

find PortNo and ServiceName

C:\oraclexe\app\oracle\product\11.2.0\server\network\ADMIN

PortNo : 1521

ServiceName : XE

==========

*imp

JDBC API:

=>'java.sql' package is JDBC API and which provide the 'classes and Interfaces' used in JDBC application development.

=>'java.sql.Connection' interface is the root of JDBC API.

=>The following are some important methods of 'Connection' interface:

1.createStatement()

- this method will create implementation object Statement interface.

2.prepareStatement()

3.prepareCall()

4.setAutoCommit()

5.getAutoCommit()

6.setSavepoint()

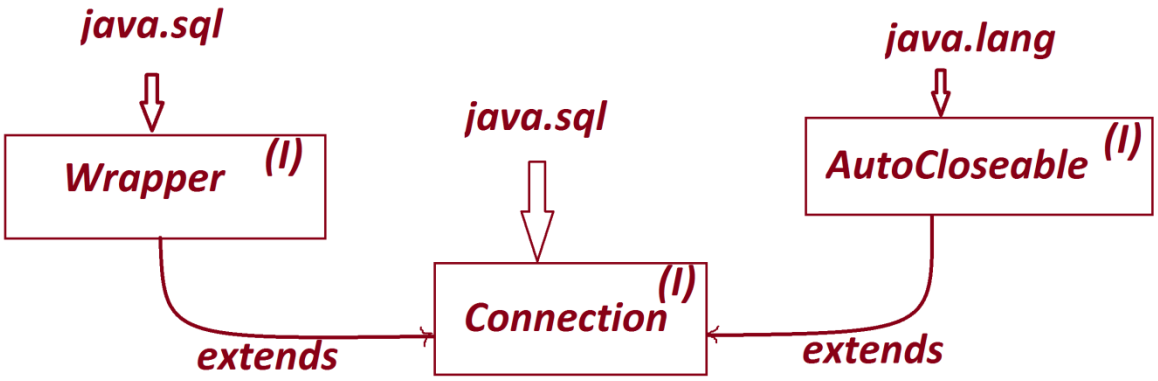
7.removeSavepoint()

8.commit()

9.rollback()

10.close() - this method will disconnect DBProduct from JavaProgram.

Hierarchy of 'Connection' interface:



==========

Dt : 2/5/2022

Note:

=>we use getConnection() method from 'java.sql.DriverManager' class

to create the implementation object of 'Connection' interface.

Method Signature of getConnection():

```
public static java.sql.Connection getConnection
(java.lang.String, java.lang.String, java.lang.String)
                throws java.sql.SQLException;
```

syntax:

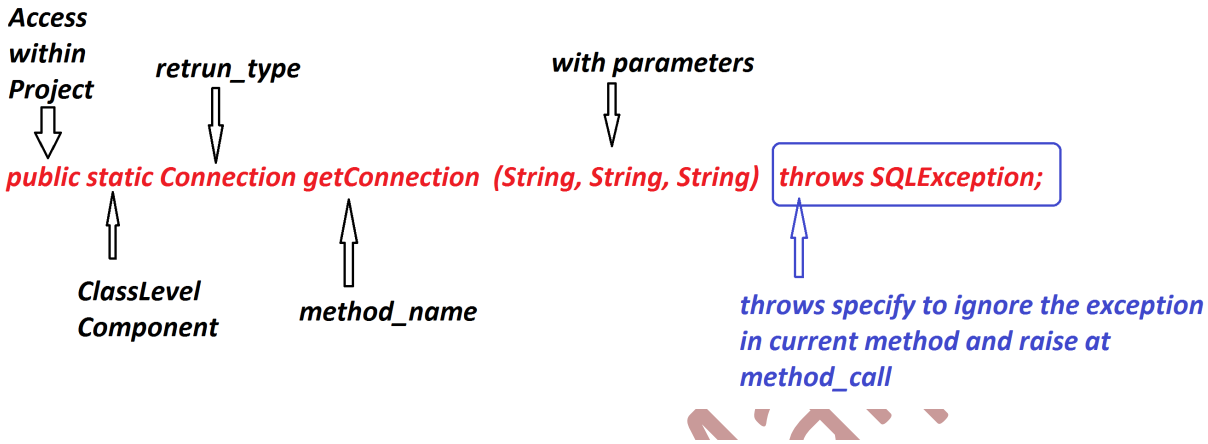
```
Connection con = DriverManager.getConnection
                ("DB-url","uname","pword");
```

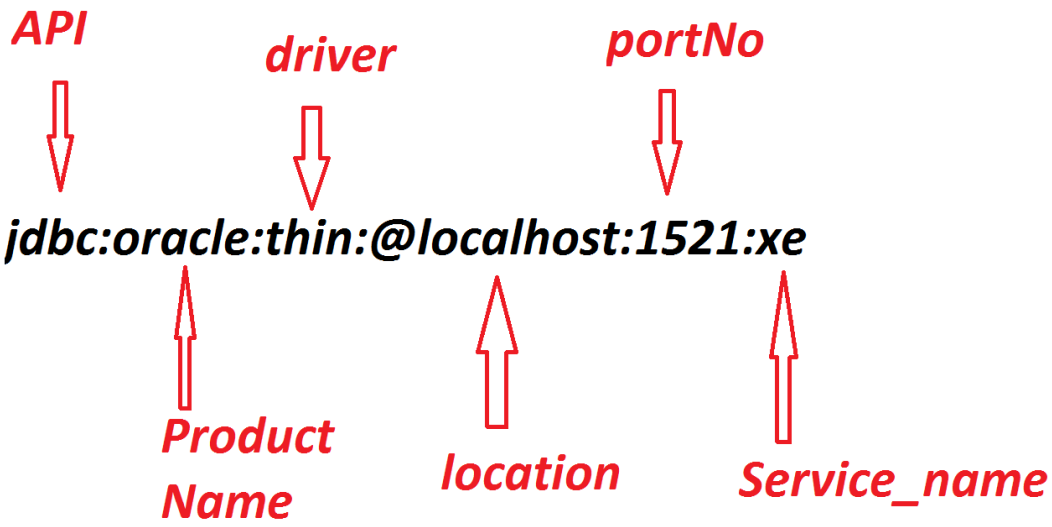
DB-URL => jdbc:oracle:thin:@localhost:1521:xe

uname => system

pword => manager

Diagram:





==========

**imp*

Types of JDBC-statements:

=>JDBC statements are used to perform operations on DataBase product.

=>These JDBC statements are categorized into three types:

1.Statement

2.PreparedStatement

3.CallableStatement

1.Statement:

=>'Statement' is an interface from java.sql package and which is used to execute normal queries without IN parameters.

=>The fllowing are some important methods from 'Statement' interface:

(i)executeQuery()

(ii)executeUpdate()

(i)executeQuery():

=>executeQuery() method is used to execute 'select' queries.

Method Signature:

public abstract java.sql.ResultSet executeQuery(java.lang.String) throws java.sql.SQLException;

(ii)executeUpdate():

=>executeUpdate() method is used to execute Non-select queries. (Create,Insert,Update and Delete)

Method Signature:

public abstract int executeUpdate(java.lang.String)

throws java.sql.SQLException;

Note:

=>we use createStatement() method from 'Connection' interface to

create the implementation object of 'Statement' interface.

Method Signature of createStatement():

public abstract java.sql.Statement createStatement()

throws java.sql.SQLException;

syntax:

Statement stm = con.createStatement();

**imp*

Creating JDBC Application using IDE Eclipse:

step-1 : open IDE Eclipse,while opening name the WorkSpace and click

'Launch'

step-2 : Create Java Project

Click on File->new->Project->Java->select 'Java Project' and click

'next'->name the project and click 'finish'

step-3 : Add DB-Jar file to JavaProject using 'Built path'

RightClick on JavaProject->Build path->Configure Build path->

Libraries->select 'classpath' and click 'Add external Jars'->

browse and select DB-Jar file->Open->Apply->Apply and Close.

step-4 : Create package in 'src'

RightClick on 'src'->new->package,name the package and click 'finish'.

step-5 : Create class in Package and type the code

RightClick on package->new->Class,name the class and click 'finish'

```
package test;
import java.sql.*;
public class DBCon1 {
        public static void main(String[] args) {
      try {
         Connection con = DriverManager.getConnection
        ("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
         Statement stm = con.createStatement();
         ResultSet rs = stm.executeQuery("select * from Product45");
         while(rs.next()) {
              System.out.println(rs.getString(1)+"\t"
                                 +rs.getString(2)+"\t"
                                 +rs.getFloat(3)+"\t"
                                 +rs.getInt(4));
         }//end of loop
```

```
            con.close();
        }catch(Exception e) {e.printStackTrace();}

            }
}
```

step-6 : Execute the program

==========

Dt : 4/5/2022

faq:

define ResultSet?

=>ResultSet is an interface from java.sql package and which hold result of select query.

=>we use executeQuery() method method to create the implementation object of 'ResultSet' interface.

=>This executeQuery() method will execute the query on DataBase product, create implementation object of ResultSet and the ResultSet object is loaded with result of selcet query.

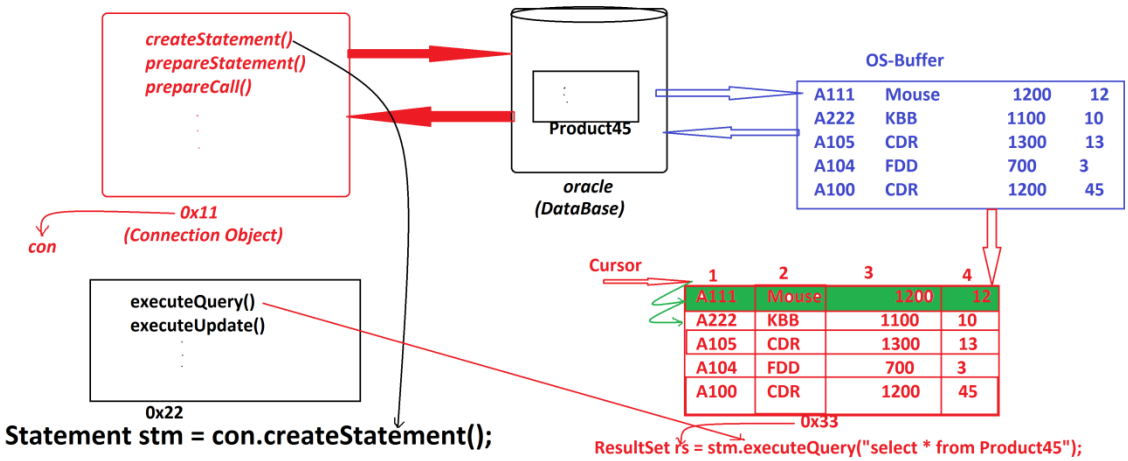
=>This executeQuery() method also generate cursor pointing before the first row.

=>we use next() method to move the cursor row-by-row and which is boolean return type.

record available means true

record Not-available means false.

Diagram:



==========

Ex_Application-2:

write JDBC application to read Product details from Console and insert into

DB Table product45.

```
package test;
import java.sql.*;
import java.util.*;
public class DBCon2 {
     public static void main(String[] args) {
         try {
             Scanner s = new Scanner(System.in);
             System.out.println("Enter the ProdCode:");
             String pCode = s.nextLine();
             System.out.println("Enter the ProdName:");
             String pName = s.nextLine();
             System.out.println("Enter the ProdPrice:");
             float pPrice = s.nextFloat();
             System.out.println("Enter the ProdQty:");
             int pQty = s.nextInt();
                 Connection con = DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:xe","system","manager");
              Statement stm = con.createStatement();
```

```
                int k = stm.executeUpdate
                ("insert into Product45 values('"+pCode+"','"+pName
                        +"',"+pPrice+","+pQty+")");
                if(k>0) {
                   System.out.println
                   ("Product Details inserted Successfully");
                }
                con.close();
              s.close();
            }catch(Exception e) {e.printStackTrace();}
        }
}
```

==========

Assignment-1:

DB Table : Book45(bcode,bname,bauthor,bprice,bqty)

Prog-1 : JDBC Application to read book details from Cosole and insert into

DB table Book45

Prog-2 : JDBC Application to display all book details

Prog-3 : JDBC Application to display book details based on book code.

==========

Assignment-2:

DB Table : Employee45(eid,ename,edesg,bsal,totsal)

Prog-1 : JDBC Application to read employee details from Cosole and insert into

DB table Employee45

Note:

Calculate totSal based on bSal using the following formula:

totSal = bSal+HRA+DA;

HRA = 93% of bSal

DA = 63% of bSal

Prog-2 : JDBC Application to display all Employee details

Prog-3 : JDBC Application to display Employee details based on eId.

==========