Dt : 17/10/2024(Day-18)

Program : DBCon14.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon14 {
	public static void main(String[] args) {
	   Scanner s = new Scanner(System.in);
	 try(s;)
	 {
	   Class.forName("oracle.jdbc.driver.OracleDriver");
	   Connection con = DriverManager.getConnection

("jdbc:oracle:thin:@localhost:1521:xe","system","tiger");
	   try(con;){
		Statement stm = con.createStatement();
		try(stm;){
			System.out.println("Enter the
NonSelect-query(create,insert,update,delete)");
			String qry = s.nextLine();
			int k = stm.executeUpdate(qry);
			System.out.println("The value in k : "+k);
			if(k>=0) {
				System.out.println("query executed
Successfully....");
			}
		}//end of try
	   }//end of try
	 }//end of try
	 catch(Exception e)
	 {
	   e.printStackTrace();
	 }
	}
}
```

o/p:

Enter the NonSelect-query(create,insert,update,delete)

create table Emp67(eid varchar2(10),ename varchar2(15),edesg varchar2(15),primary key(eid))

The value in k : 0

query executed Successfully....

Enter the NonSelect-query(create,insert,update,delete)

insert into Emp67 values('A121','Alex','SE')

The value in k : 1

query executed Successfully....

Enter the NonSelect-query(create,insert,update,delete)

update Emp67 set edesg='SE' where eid='E231'

The value in k : 1

query executed Successfully....

Enter the NonSelect-query(create,insert,update,delete)

delete from Emp67 where eid='D231'

The value in k : 1

query executed Successfully....

==========

faq:

define 'RowSet'?

=>'RowSet' is an interface from 'javax.sql' package and which is extended from

'java.sql.ResultSet' interface.

(RowSet is a Child-Interface of ResultSet)

=>RowSet-Object also will hold the result generated from Select-queries.

=>RowSet-Objects are automatically Scrollable-Objects.

=>This 'RowSet' is categorized into two types:

1.JdbcRowSet

2.CachedRowSet

1.JdbcRowSet:

=>After getting the data from database product,JdbcRowSet-object still connected to database product.

2.CachedRowSet:

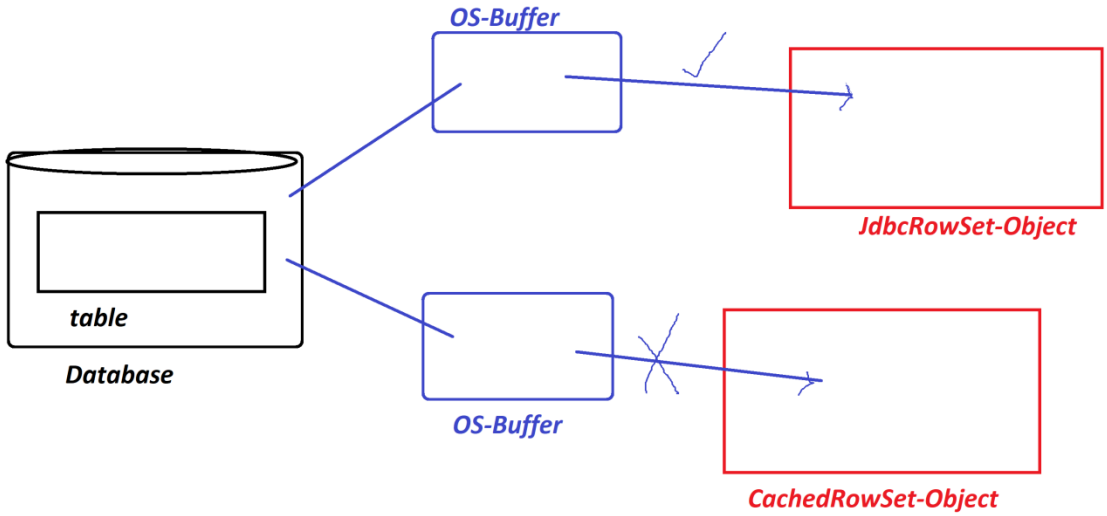
=>After getting the data from database product,CachedRowSet-object is disConnected from database product automatically.

Note:

=>In realtime,we use CachedRowset in the form of WebRowSet for getting the data from Database product.

java - represent Java-Library

javax - represent Extended-Java-Library



==========

*imp

Types of JDBC drivers:

=>These JDBC drivers are categorized into four types:

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

2.Native API driver(Type-2 driver)

3.Network Protocol driver(Type-3 driver)

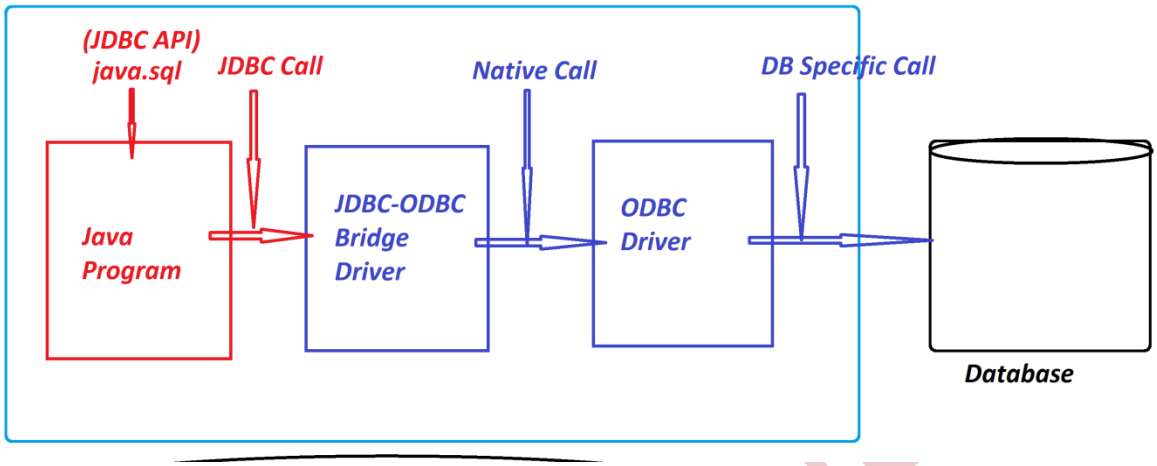
4.Thin driver(Type-4 driver)

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

=>Type-1 diver will take the support of ODBC-driver to establish connection to Database product.

=>In Type-1 driver,internally JDBC-Call converted into Native call,and this Native call is converted into DB Specific call for connection.

Diagram:



DisAdvantage:

=>Type-1 driver internally uses more conersions,and which waste the execution time and degrades the performance of an application.

Note:

=>From Java8 version(2014) onwards Type-1 driver support is not available.

faq:

define ODBC driver?

=>ODBC stands for 'Open DataBase Connectivity' and which is used to communicate with any type database.

=>ODBC-driver internally uses c/c++ code and which is PlatForm dependent driver.

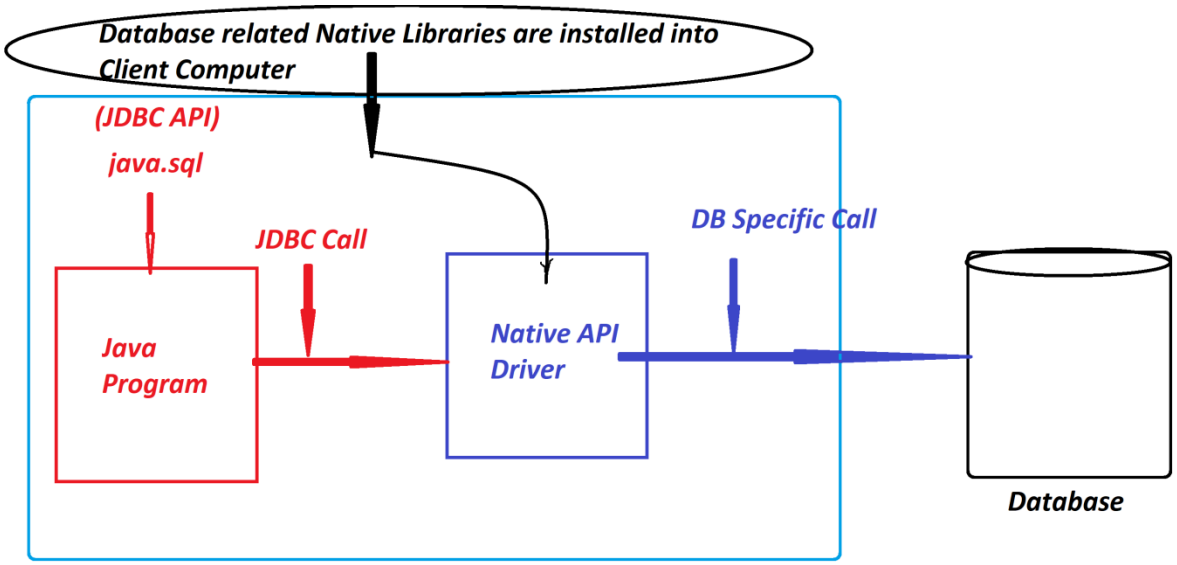
2.Native API driver(Type-2 driver):

=>Type-2 driver will take the support of 'Database related Native Libraries' to

establish connection to Database product.

=>when we want to use Type-2 driver,the Client Computer must be installed with 'Database related Native libraries'

Diagram:



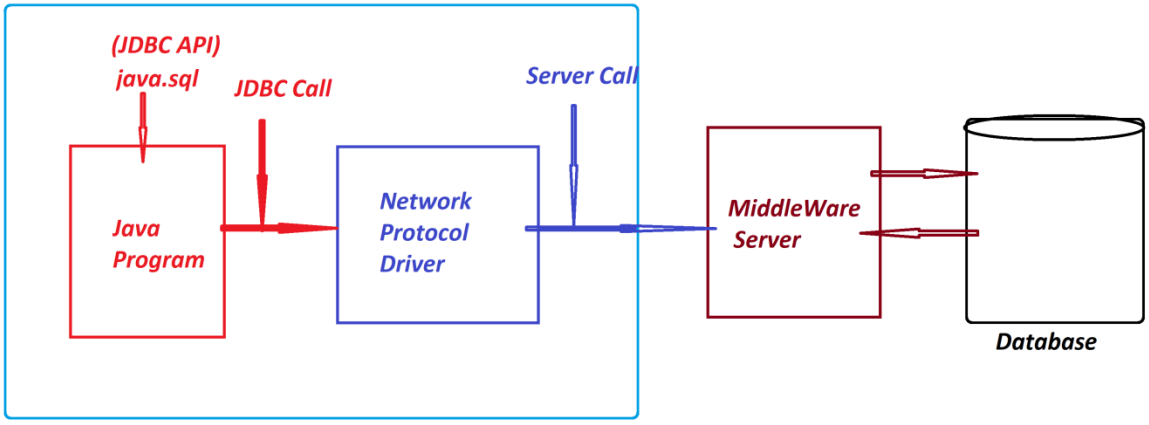
DisAdvantage:

=>when we use Type-2 driver,the application will become Database dependent and which is not preferable in realtime.

3.Network Protocol driver(Type-3 driver):

=>Type-3 driver will take the support of intermediate Middle-Ware servers to communicate with database product.

Diagram:



DisAdvantage:

=>In Type-3 driver Network related code and components are involved,and execution time increases and degrades the performance of an application.

4.Thin driver(Type-4 driver):

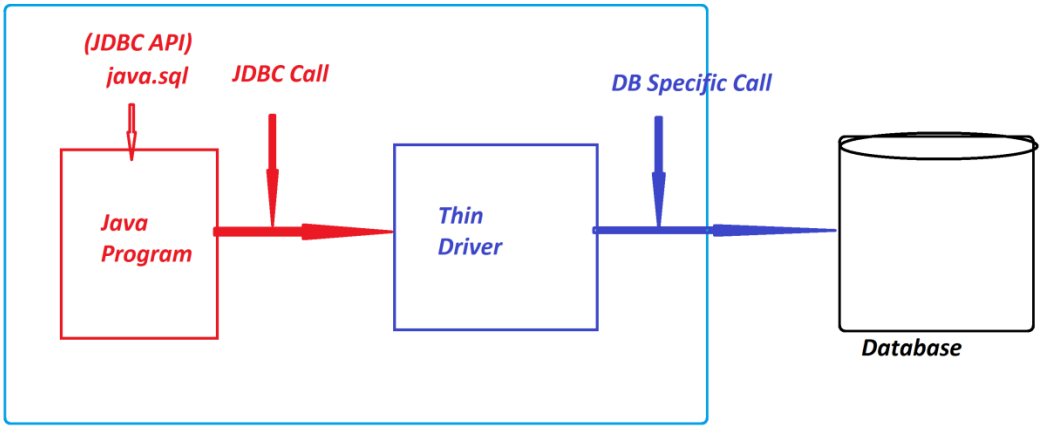
=>Type-4 driver will take the support of JDBC-Network-Protocol to establish Connection to Database product

=>In Type-4 driver(Thin driver) the JDBC-call converted directly into DB Specific call for connection.

=>Type-4 driver is pure Java driver and PlatForm independent driver.

=>This Type-4 driver is HighPerformance driver.

Diagram:



==========