Dt : 13/2/2025

faq:

define request?

=>The query which is raised by the user through WebBrowser is known as 'request'

=>'request' is categorized into two types:

1.POST request

2.GET request

1.POST request:

=>The request which is raised to send the data to server is known as POST request.

=>Through POST request we can send all types of data,which means we can send Text,Audio, Video,Image and Animation.

=>Through POST request we can send UnLimited data.

=>The data in POST request is secure,because the data is encapsulated into the body of HTTP.

=>To raise POST request,we write method="POST" in <form> tag.

syntax:

```
<form action="url" method="POST">
    ...
</form>
```

=>we use doPost()-method from 'HttpServlet' to accept POST request.

Method Signature of doPost():

```
protected void doPost(jakarta.servlet.http.HttpServletRequest,
 jakarta.servlet.http.HttpServletResponse) throws jakarta.servlet.ServletException,
            java.io.IOException;
```

2.GET request:

=>The request which is raised to get data from Server is known as GET request.

=>Through GET request we can send only Text data.

=>Through GET request we can send max up to 10KB data(Limited data)

=>The data in GET request is not secure,because the data is dispayed in address-bar directly.

=>We use the following syntaxes to raise GET request:

syntax-1 : we use method="GET" in <form> tag

```
<form action="url" method="GET">
 ...
</form>
```

syntax-2 : If we construct <form> tag without any method specificaation will raise GET request

```
<form action="url">
 ...
</form>
```

syntax-3 : GET request is raised through hyperlinks.

syntax-4 : The GET request can be raised through address-bar,by using servlet-url-pattern

=>We use doGet()-method from 'HttpServlet' to accept GET request.

Method Signature of doGet():

protected void doGet(jakarta.servlet.http.HttpServletRequest,

jakarta.servlet.http.HttpServletResponse) throws jakarta.servlet.ServletException,

java.io.IOException;

---

faq:

define Session?

=>The time interval b/w User-Login to User-Logout is known as Session.

faq:

define Session Management?

=>The process of recording the state-of-user and tracking the used in session,is known as Session Management.

Types of Session Tracking Techniques used in Session Management:

1.HttpSession

2.URL re-write

3.Hidden Form Fields

4.Cookie

1.HttpSession:

=>'HttpSession' is an interface from jakarta.servlet.http package and,which is instantiated and used in Session Tracking process.

=>The following are some important methods of HttpSession:

public abstract void setAttribute(java.lang.String, java.lang.Object);

public abstract java.lang.Object getAttribute(java.lang.String);

public abstract void removeAttribute(java.lang.String);

public abstract java.util.Enumeration<java.lang.String> getAttributeNames();

public abstract void invalidate();

public abstract jakarta.servlet.ServletContext getServletContext();

Note:

=>we use getSession()-method from 'HttpServletRequest' to create implementation Object for HttpSession-Interface.

Method Signatures of getSession():

public abstract jakarta.servlet.http.HttpSession getSession();

public abstract jakarta.servlet.http.HttpSession getSession(boolean);

==========

Application : ProductStore

DBTables:

Product70(code,name,price,qty)

Primary Key : code

Admin70(uname,pword,fname,lname,city,mid,phno)

Primary Key : uname and pword

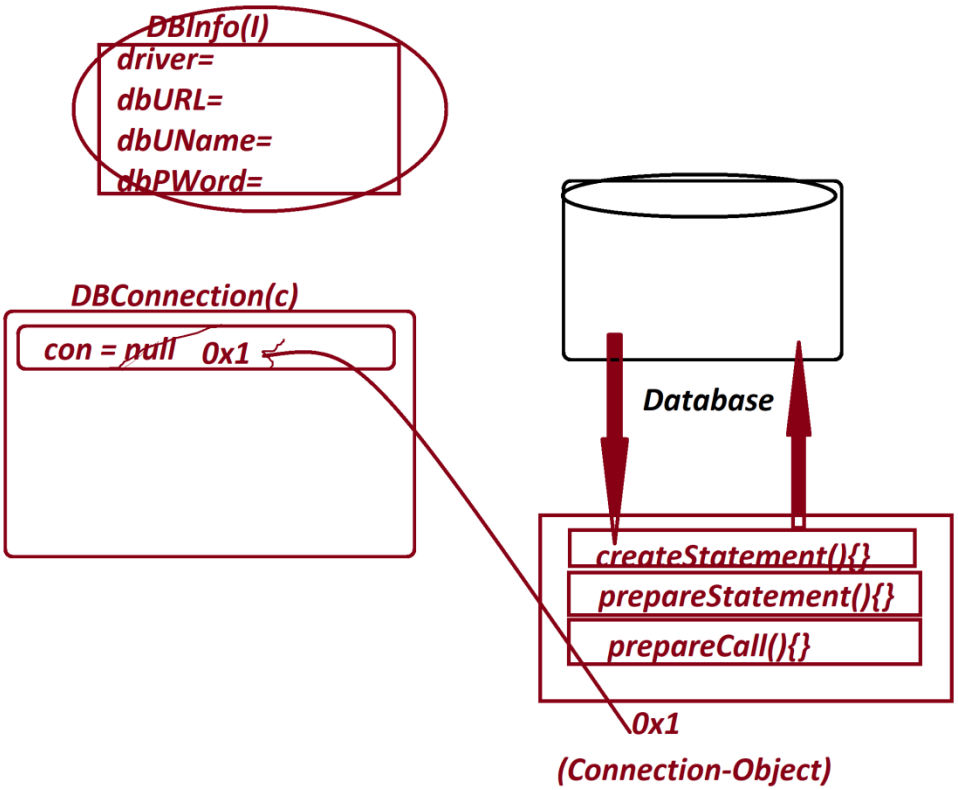
create table Admin70(uname varchar2(15),pword varchar2(15),fname varchar2(15),

lname varchar2(15),city varchar2(15),mid varchar2(25),phno number(15),

primary key(uname,pword));

insert into Admin70 values('nit.v','mzu672','V','M','Hyd','v@gmail.com',9898981234);

Layout:



DBInfo.java(Interface)

```
package test;
public interface DBInfo
{
   public static final String driver="oracle.jdbc.driver.OracleDriver";
   public static final String dbURL="jdbc:oracle:thin:@localhost:1521:xe";
   public static final String dbUName="system";
   public static final String dbPWord="tiger";
}
```

DBConnection.java

```
package test;
import java.sql.*;
public class DBConnection
```

```
{
	private static Connection con = null;
	private DBConnection() {}
	static
	{
	  try {
		Class.forName(DBInfo.driver);
		con = DriverManager.getConnection
				(DBInfo.dbUName,DBInfo.dbUName,DBInfo.dbPWord);
	  }catch(Exception e) {
		e.printStackTrace();
	  }
	}//end of block
	public static Connection getCon()
	{
	  return con;
	}
}
```