**Start Date-28/11/12**

**J2EE(JAVA 2 ENTERPRISE EDITION)**

1)internet is the global internet that we can access any information all over the global.but intranet is the local internet that can access within the local place.like within the company.

2)gmail,facebook,flipcart etc all are web application that works on internet.

3)all web application works on 2-tier or multiple tier.

4)when clients fetch the data directly from database server i.e called 2-tier.the most disadvantage is maintenance is very high.e.g-ATM machine,Trading,the times of india.

5)when clients send the request to web server then web server interact with database & give the response back to the client i.e. called 3-tier.e.g.gmail,facebook,flipcart etc.

6)enterprise means more & more functionality.

7)IDE-Interactive development environment.

8)static pages is the constant page that will be same for all users.but dynamic pages will keep change with respect to user.

10) **HTTP(Hyper Text Transfer Protocol)**-

-this is a defined process of how to transfer information between a web browser and a web server.

-all web browser & web server follow this process.

11)**HTML(Hyper Text Markup Language)-**

-this is a language used in web pages to format text,images & page layout.

12) **SERVER-**

**-**are those machines that provide services to other machines(like web server or FTP server).

13) **CLIENT-**

**-**are those machines that are used to connect to those services.

14) **WEB BROWSER**-

-Is a piece of software that is used to interpret the information found in an HTML document and display the content based on the HTML tags.e.g.microsoft internet explorer.e.g.http server

15) **WEB SERVER**-

-a web server is a piece of software that manages web pages and makes them available on client browser-via local network or over the internet.

16) **IP ADDRESS**

**-**each machine on the internet is assigned a unique address i.e. called IP address.

-these are 32-bit number,normally expressed as four “octets” in a “dotted decimal number”.

-a typical IP address look like this:218.29.65.143

-the four numbers of IP address are called octets because they can have values between 0 to 255,which is 2^8 possibilities per octet.

**Introduction of J2EE(JAVA 2 ENTERPRISE EDITION)**

1)**Java 2 platform Enterprise Edition(J2EE)** is a platform independent,java centric environment for developing web based enterprise application.

|  |
| --- |
| HIBERNATE,SPRINGS,STRUTS etc FRAMEWORK |
| JSP  J2EE SERVLET |
| CORE JAVA |

2)JDK is platform independent but JRE is platform dependent.

3)J2EE platform consists of a set of services ,API and protocols that help us for developing multi tier web based enterprise application.

**J2EE 3-TIER ARCHITECTURE**

Tier-1 Tier-2 Tier-3

**MAJOR ELEMENTS OF J2EE**

**JDBC(Java Database Connectivity)-**

1)JDBC is an API that provides connectivity between databases and J2EE platforms.

2)if we have an application and it has a DB.we have to interact with DB to read,update,delete and insert information of data.

3)there are many ways to interact with DB and JDBC is one way.

**WHY JDBC?**

For reasons why we need JDBC are-

1)JDBC is not DB dependent and also we can interact with multiple DB simultaneously.

2)JDBC support stored procedure.

3)we can achieve high performance by using JDBC(e.g. PreparedStatement).

4)since it is builtup on java eventually inherits the advantages of JAVA.

**NECESSARY STEPS TO WORK WITH JDBC**

**1)**load the JDBC driver.

2)create the **Connection** object using the **driver manager**.

3)create **Statement** object using the **Connection** object.

4)issue SQL statement using the **Statement** object and **ResultSet** should be returned.

5)Iterate **ResultSet** through the rows returned and display the data.

6)close all the JDBC object.

->MySQL URL syntax

jdbc:mysql://[*host*][,*failoverhost*...] [:*port*]/[*database*] [?*propertyName1*][=*propertyValue1*] [&*propertyName2*][=*propertyValue2*]...

->default value of host & port are 127.0.0.1 & 3306

String ConnectionURL=”jdbc:mysql://localhost:3306/movies?user=scott&password=tiger”;

Class.forName(“jdbc:mysql.Driver”);

Connection con=DriverManager.getConnection(connectionURL);

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(“select \* from movies”);

While(rs.next())

{

System.out.println(“Name=”+rs.getString(“moviename”)+”Date=”+rs.getString(“releasedate”);

}

**DRIVER**

1)a driver is a additional software component required by JDBC to interact with database.

2)Drivers are DB specific(\*).

3)the drivers opens the connection to the database and transfers DB query and results between java program and database.

4)Two ways to load the driver:-

-Class.forName(“com.sql.jdbc.Driver”);

-Driver driver=new com.sql.jdbc.Driver;

DriverManager.registerDriver(driver);

**Driver Type**

There are four diff. types of drivers.

**1)Type1 Driver:JDBC-ODBC Bridge:-**

->this drivers influence the ODBC driver to connect to the database.this drivers converts JDBC method calls into ODBC function calls.

->this driver is platform dependent.

**Disadvantages**

->Performance Overhead-since a calls go through the JDBC to ODBC driver,then to the native DB connectivity interface.

->these driver is not suitable for high transaction environment.

->the ODBC driver needs to install to the client machine.

DATABASE

JDBC

ODBC DRIVER

ODBC BRIDGE

TYPE1 DRIVER

**2)Type2 Driver:Native-API Driver:-**

**->**this driver uses the client side libraryof the database.

->it converts the JDBC method calls to native calls of database using this client side library.

->it is platform dependent.

**Disadvantages**

->the vendor client library needs to be installed at the client machine.

->not all databases have client side library.

DATABASE

TYPE2DRIVER

JDBC

CLIENT LIBRARY

**3)Type3 Driver:Network-Protocol Driver:-**

->it’s a pure java driver makes use of middle ware server between the calling program & database.

->this middle ware server converts JDBC calls either directly or indirectly to vendor specific DB calls.

->it is platform independent.

->it is used for multiple database.it depends on the number of database,the middle ware server has been configured to support.

**Disadvantages**

->the middle ware server may result in additional latency(major of time delay experience in a system).

DATABASE

DATABASE

MIDDLEWARE SERVER LIBRARY

TYPE3DRIVER

JDBC

DATABASE

**4)Type4 Driver:Native-Protocol Driver:-**

->this driver also known as the directive database pure java driver.

->this driver implementation converts JDBC calls directly to a DB calls.

->it is platform independent.

->it provides better performance than other driver because it doesn’t have the overhead or conversion of calls into ODBC or database APIs calls.

**Disadvantages**

->it is database dependent.

DATABASE

TYPE4DRIVER

JDBC

**JDBC Driver Summary:-**

DATABASE



ODBC BRIDGE

ODBC DRIVER

TYPE1 DRIVER

JDBC

CLIENT LIBRARY

TYPE2 DRIVER

MIDDLEWARE SERVER

TYPE4 DRIVER

TYPE3 DRIVER

TYPE1 DRIVER 1.client installation

TYPE2 DRIVER 2.platform dependent.

3.performance overhead

TYPE3 DRIVER 1.100% pure java----------interact with multi DB.

TYPE4 DRIVER 2.platform independent.-----------DB dependent.

->for shortcut press ctrl+alt+L

**URL(uniform Resource Locator)-**

these are the addresses that we enter into our web browser to connect to a website.it contains four parts.

->**Protocol-**

protocol is the string of character that we use before the hostname.eg.http,ftp,telnet etcthey are separated by colon & two forward slashes(://).

-these protocols tell our browser what type of service to use when we connect with the web browser to the hostname.

-by default,web browser will take http protocol.

**->HOSTNAME**-

-hostname is the address,from where we want to access any data.eg.www.google.com.

**->PORT NUMBER**-

-this is a number that we can append with the hostname through colon(:).

-by default its taking port no 80.e.eg. <http://www.google.com:80>.

**->PATH**-

-This is the path from where we want to access any file.e.g.

http :// [www.example.com](http://www.example.com) :80 /chapters/chapter1.html

protocol hostname portno path

jdbc:mysql :// localhost :3306 /class ?user=root&password=root

**MySQL-**jdbc:mysql://localhost:3306/class?user=root&password=root

**Oracle-** jdbc:oracle:thin://localhost:1158","scott","tiger

**MycrosoftSqlServer-**jdbc:Microsoft:sqlserver://myserver:1433;DatabaseName=mydb;user=root;password=root;

**DriverManager**

1)as the name implies,it manages the driver.since jdbc is not DB dependent and it can interact with multiple DB simultaneously.using suitable DB drivers & its corresponds URL.DriverManager will set the DB connectionfrom one or more database.

2)Driver manager has static getConnection method which attempts to establish a connection to the given DB url.the driver manager attempts to select an appropriate driver from the set of registered JDBC driver.

3)driverManager has three overloaded getConnection method.

DriverManager.getConnection(“<url>”);

DriverManager.getConnection(“<url>”,“<userID>”,“<password>”);

DriverManager.getConnection(“<url>”,properties,properties);

🡪properties file contains-<key>=<value>

🡪metadata-data on data(extra information).

**Connection**

1)connection represents the session to a specific database.

2)within the context of the connection,sql statements are executed and results are returned.

3)we can have multiple connection to a DB.

4)it also provides information about databases,tables,fields etc(metadata).

5)it has method to deal with transactions.

**JDBC Statements**

1)JDBC statement send sql query and receive the data from database.

2)JDBC statements also has methods that help to bridge datatype instances between java and DB datatypes.

3)there are three different types of JDBC statement.

->Statement

->PreparedStatement

->CollableStatement

\*\*once we have created a statement object(any one the above type) then we can use to execute a SQL statement with one of its four executes methods.

🡪**ResultSet executeQuery()**:-returns a ResultSet object .this method is used to execute SQL select statement.

🡪**int executeUpdate()**:-returns no of rows affected by the execution of SQL statement.this method is used to execute SQL insert,update & delete statement.

🡪**boolean execute()**:-

-this method executes any given SQL statements(both DDL & DML commands).

-if we use this method then we must use the method getResultSet() or getUpdateCount() to retrieve the result.

-this method returns true if the result is a ResultSet object;false if the an update count.

🡪**int[] executeBatch():-**this method groups the related SQL statement into a batch and submit them with one call to database.if all command execute successfully returns an array of update counts.

🡪**void addBatch():-**we add individual statement or prepared statement or callable statement to batch using this method.

🡪**void clearBatch():-**this method removes all the statement present in the batch.however we can’t selectively choose which statement to remove from batch.

**STATEMENT**

1)statements are used when we want to execute static SQL statements at run time.

2)the statement can’t accept query parameter at runtime instant they are used with already set parameters.

*stmt*=*con*.createStatement();

ResultSet rs=stmt.executeQuery("select \* from student");

while(rs.next())

{

System.out.println("fname:"+rs.getString("fname")+"lname:"+rs.getString("lname")+"usn:"+rs.getString("usn"));

}

}

*stmt*=*con*.createStatement();

**int** count=*stmt*.executeUpdate("create table emp3(ename varchar(10),deptno int(2),sal int(10))");

System.*out*.println(count+"rows updated");

boolean isExecuted=stmt.execute("create table emp1(ename varchar(10),deptno int(2),sal int(10))");

boolean isExecuted1=stmt.execute("insert into emp values('vikash',10,20000)");

boolean isExecuted=stmt.execute("select \*from student");

if(isExecuted)

{

System.out.println("ResultSet has ResultSet object");

rs=stmt.getResultSet();

while(rs.next())

{

System.out.println("fame:"+rs.getString("fname")+"lname:"+rs.getString("lname")+"usn:"+rs.getString("usn"));

}

}

else

{

System.out.println("ResultSet has updated count");

int count1=stmt.getUpdateCount();

System.out.println(count1+"table created");

}

**JAVA BEAN**

1)it is a final class.it should have public default constructor.

2)it should have private variable & should have getter & setter method t access private variable.

3)should not have any compotentional logic & method.

**Use**

1)to hold the data among the program.

->ArrayList<Student> list=new Arraylist<Student>();

**PREPARED STATEMENT**

1)preared statement are used when we want to execute the same SQL statements many times.

2)they are also used in the situation where query parameters are set at runtime(dynamic SQL statements).

3)the prepared statement uses the ‘?’.

3)it is performance oriented.

4)they are also known as precompiled statement and they must be used with query parameters.

PreparedStatement pstmt=con.prepareStatement("select \* from student where fname=?");

pstmt.setString(1,args[0]);

ResultSet rs=pstmt.executeQuery();

5)Question marks are called query parameters are place folder or bind variables.

**HOW THE PREPARED STATEMENT INCREASE THE PERFORMANCE**

**Background of DB Query Execution**

1)when a database receives the statements, the database using parses the statement and looks for syntax error.

2)when the statement is passed,DB starts preparing the execution plan i.e. it is figure out the most efficient way to execute the statement.this is compotenally quite expensive.once the execution plan is created then using the plan DB engine executes the statements.

**Statement Catches**

1)Databases are designed to do statement catch.the catch uses the statement itself as a key and the execution plan as a value corresponding to this key.

2)this allows database engine to receive the plan per statements that have been executed previously.for example if we sent database statement such as “select a,b from xyz where c=?”;then computed execution plan will catch.if we sent the same statement later the DB can use previous execution plan.it is saving server resources along with the time for generating execution plan.

3)however the entire statement is the key.for example-if we sent the statement same as above.it wouldnot find the execution plan.this is because c=3,is different from catch plan c=2;by using parameterized prepared statement we can improve the efficiency of the database and on application.

**CALLABLE STATEMENT**

1)it is used to execute stored procedure.

**Stored Procedure**

1)it is a group of SQL commands that perform particular task that it is a name implies.it is stored in database.

/\*

\* STORED PROCEDURE

\*

delimiter $$

create proedure getAllStudentInfo()

begin

select \* from student;

end $$

delimiter;

call getAllStudentInfo();

\*/

/\*

\* STORED PROCEDURE

\*

delimiter $$

create proedure getAllStudentInfo(in registerNo int)

begin

select \* from student where regNo=registerNo;

end $$

delimiter;

call getAllStudentInfo(1);

\*/

**ResultSet**

1)it is an order set of rules produces by the DB query.

2)resultsets are produced by executing query method or certain metadata method calls.

3)once the ResultSet is produced data from resultset can be extracted as follows.

-move to desired row by calling necessary methods e.g.next(),last() etc

-retrieve the data column value using getXXX(<column\_name>);getXXX(<column\_no\_in\_SQLStatement>);

**Why we need to close necessary JDBC object**

1)Connection,Statement,ResultSet are often tied to OS resources such as memory.

2)in case of connection further resources are tied to DB server.it is vital to close any JDBC object as soon as it played it prt.it should be relied upon.

3)forgotten to close this properly,results in suspicious errors & misbehavior.

**TRANSACTION**

1)Trasaction is a set of one or more statements that are executed as a unit,so either all the statements are executed or non of the statements are executed.

2)following steps are followed to handled to transaction in JDBC.

->begin the transaction decipline auto commit mode(con.setAutoCommit(false)).

->execute one or more data manipulation language(DML-insert,delete,update).

->if no error occur then commit the transaction(con.commit()).

->if error occurs then rollback the transaction(con.rollback()).

**BATCH PROCESSING**

1)it allows us to do different sql statements into a batch and submit them with one call to DB.it is used with DML(insert,update,delete).

2)when we send several sql statements to the database at once it reduces the amount of communication overhead and increase the performance.

3)steps to do batch processing.

->add individual statements or prepared statements or callable statements to batch using **addBatch()** method.

->using **executeBatch() ,**start the execution of the batch.it returns the array of integer and each elements of the array represents the count for respective DML commands.

->for example,if there are 20 operations in a batch,if first 13 get executed successfully and 14th has generated an exeception then update count array will have 13 elements.then we can either commit or rollback the successful operations in this situation as we preferred.

->**use clearBatch()** method to remove all statements present in the batch.however we can’t selectively choose which statement to remove from batch.

**CONNECTION POOLING**

1)a connection pool is a cache of database connection objects.these objects represents physical database connections that can be used by in application to connect to a database.

2)connection pool promote the reuse of connection object and reduce the no of times that connection objects are created.**connection pool significantly improve performance of database intensive application because creating connection object is costly both in terms of type and resources.**

3)with connection pooling,the pool of connection are created at the time of application startup are when first DB connection request are made.hence application waits less time to get the connection.

4)connection pools often provide properties that are used to optimized pools performance.the properties control the behavior of connection pool.it contain information such as minimum & maximum no of connection allowed in the pool,userID & password etc to get connection.

5)always connection pool has been resided in firewall.firewall is a secure level,without permission we can’t access.

6)using singleton design pattern,we can create the object once through private constructor.

->TAR file can download for unix & linux.(.sh) is shell file is used for unix.

Singleton

1. Provide a default Private constructor
2. Create a Method for getting the reference to the Singleton Object
3. Make the Access method Synchronized to prevent Thread Problems.
4. Override the Object clone method to prevent cloning