SVKM’s NMIMS University

Mukesh Patel School of Technology Management & Engineering

**MCA SEMESTER III**

**SUBJECT: Web Technologies Practical Assignment: 8**

# Part A (To be referred by students)

**Topic: Implementation of Database Connectivity using JSP Aim:** To establish database connection with JSP

# Outcome: After successful completion of this practical students will be able to

1. Understand how to connect Database with JSP page

# Theory:

**Steps for JDBC**

# Create a Connection

<% Connection conn=null; %>

# Load JDBC Driver

<% Class.forName("com.mysql.jdbc.Driver").newInstance(); %>

# Open database Connection

<% conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root", ""); %>

# Create Statement/PreparedStatement for executing query

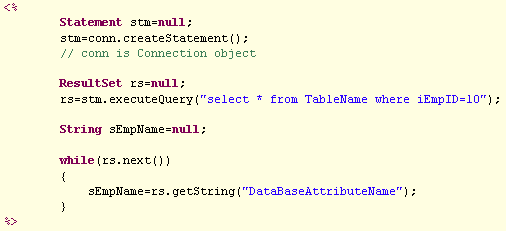
<%

Statement stm=null; stm=conn.createStatement();

stm.executeUpdate("insert into tableName values('FieldName1','FieldName2')");

%>

# Create ResultSet and Fetch Data from Resultset



1. **Close Resutlt set, Statement and connection**

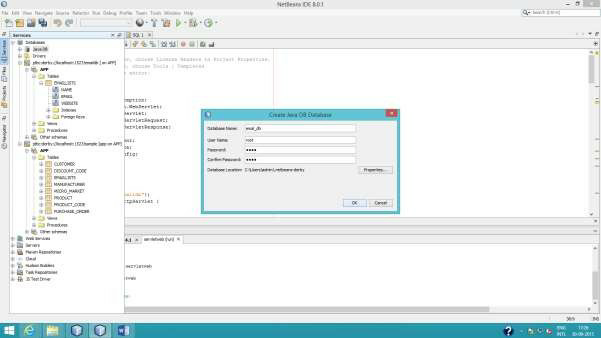
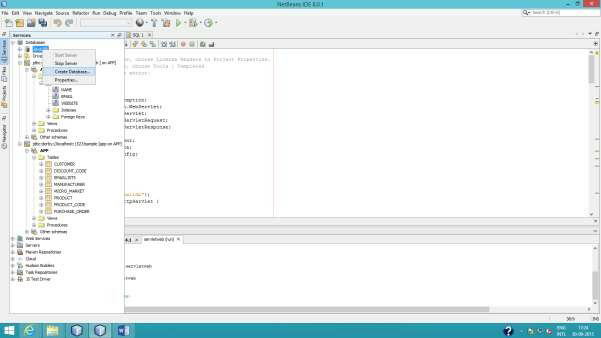
# <% rs.close(); rs = null; stmt.close(); stmt = null;

**conn.close(); // Return to connection pool conn = null; %>**

# Example of creating Database connection with derby database using Net Beans.

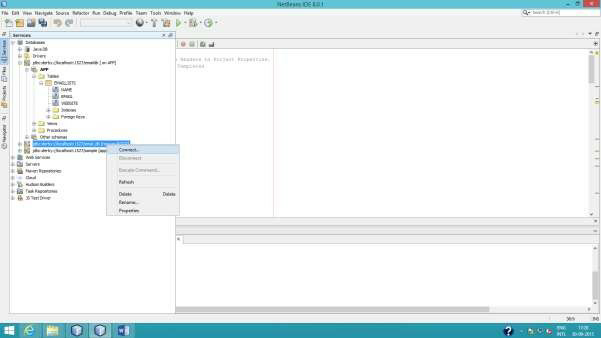
Steps for Database connection with derby database

1. Create a new database using server tab of netbeans

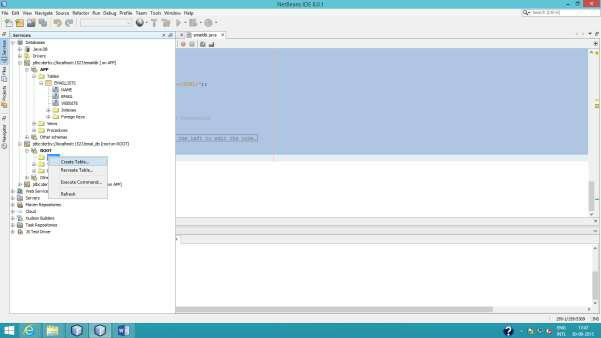


Give database name and user name and password and click on ok

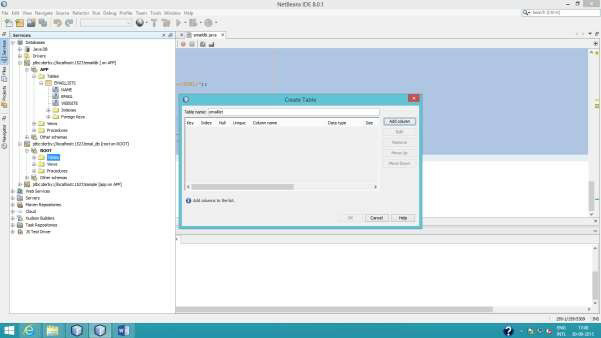
1. Right click the database which you have created and select connect

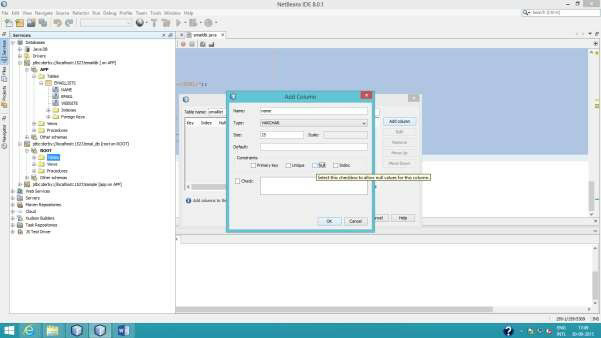


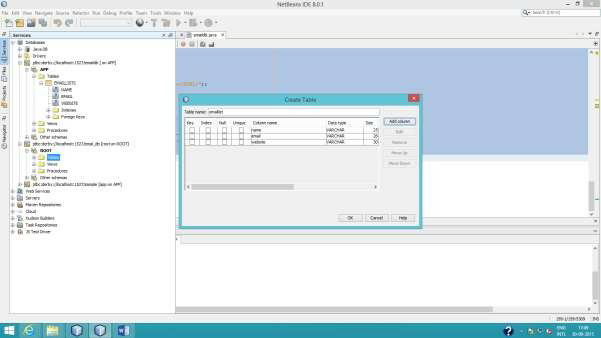
Create table emaillists for emaildb database



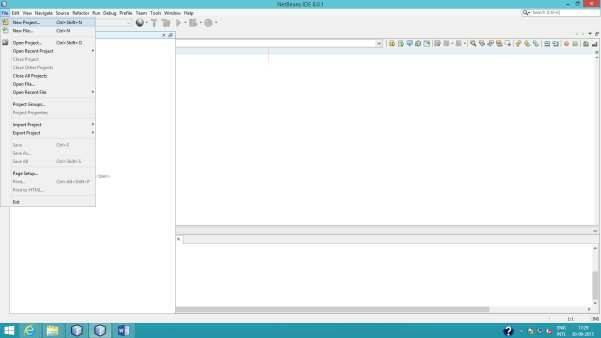
Add proper columns for table



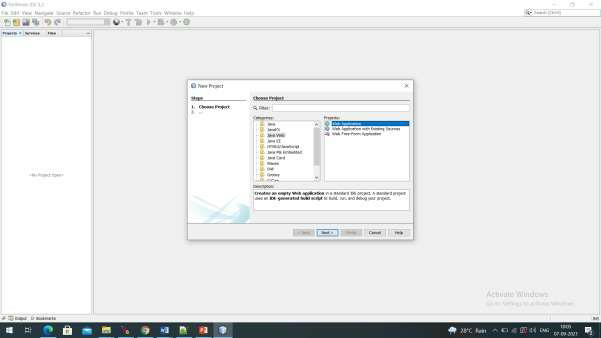




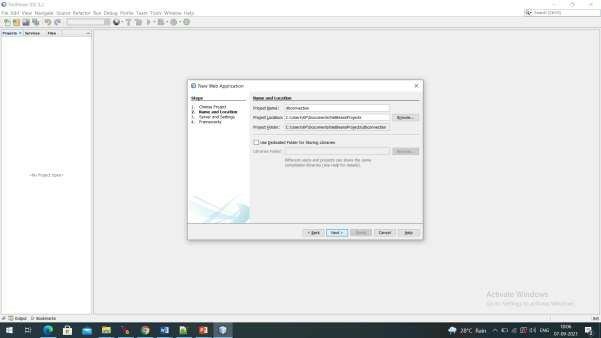
1. Go to project tab in netbeans create new project



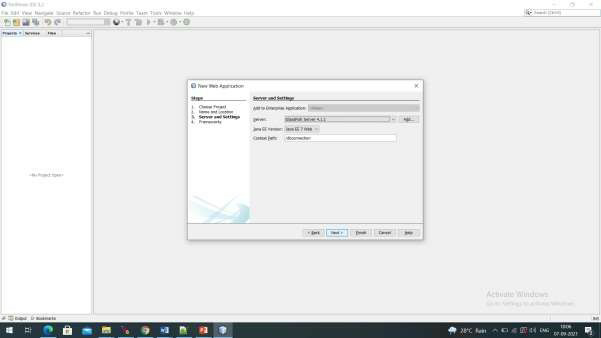
Select java web-> web application



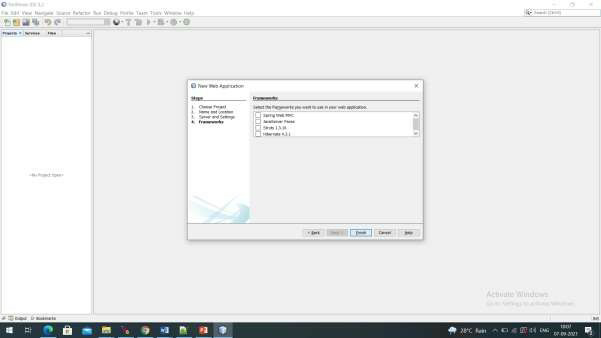
Give the name as dbconnection



Click on next select the sever

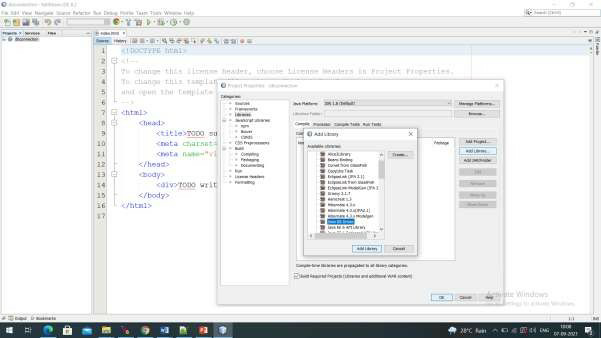


Click on next and then finish



You can easily achieve that adding the JavaDB library to your project:

* Right-click on the project
* Properties -> Libraries
* Add Library -> JavaDB driver



# Dbconnection.html

JSP with JDBC connection

Name :

Email : Website :

Options : Add Display Remove Update

Submit

# Code:

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form name=emailaccess method=get action="emaildb.jsp">

<h1>JSP with JDBC connection</h1>

<table border="0">

<tr><td>Name : </td><td><input type="text" name="my\_name" size="20"></td></tr>

<tr><td>Email : </td><td><input type="text" name="my\_email" size="40"></td></tr>

<tr><td>Website : </td><td><input type="text" name="my\_website" size="40"></td></tr>

<tr><td>Options : </td><td><input type="radio" name="my\_choice" value="Add">Add</td></tr>

<tr><td></td><td><input type="radio" name="my\_choice" value="Display" checked>Display</td></tr>

<tr><td></td><td><input type="radio" name="my\_choice" value="Remove">Remove</td></tr>

<tr><td></td><td><input type="radio" name="my\_choice" value="Update">Update</td></tr>

<tr><td colspan="2"><input type="submit" name="callemaildb"></td></tr>

</table>

</form>

</body>

</html>

# Emaildb.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<body>

<%@ page import="java.sql.\*" %>

<h1>JSP Database Example</h1>

<%

Connection conn = null; Statement stmt = null; ResultSet rs = null;

try {

String dbURL = "jdbc:derby://localhost:1527/emaildb;create=true;user=app;password=app";

Class.forName("org.apache.derby.jdbc.ClientDriver").newInstance(); conn= DriverManager.getConnection(dbURL);

stmt = conn.createStatement();

%>

<table border =1>

<tr>

<td>Name</td><td>Email address</td>

<td>Website</td>

</tr>

<%

String action=request.getParameter("my\_choice");

if(action.equals("Remove"))

{

String nm=request.getParameter("my\_name");

String template = "Delete from emaillists where name = ?"; PreparedStatement statement = conn.prepareStatement(template); statement.setString(1,nm);

statement.execute();

}

if(action.equals("Add"))

{

String nm=request.getParameter("my\_name"); String em=request.getParameter("my\_email"); String wb=request.getParameter("my\_website");

String template = "Insert into emaillists(name,email,website) values (?,?,?)"; PreparedStatement statement = conn.prepareStatement(template); statement.setString(1,nm);

statement.setString(2,em); statement.setString(3,wb); statement.execute();

}

if(action.equals("Update"))

{

String nm=request.getParameter("my\_name"); String em=request.getParameter("my\_email"); String wb=request.getParameter("my\_website");

String template = "Update emaillists SET email=?, website=? where name=?"; PreparedStatement statement = conn.prepareStatement(template); statement.setString(1,em); statement.setString(2,wb); statement.setString(3,nm);

statement.execute();

}%>

<% // You need to edit this query

rs = stmt.executeQuery("SELECT \* FROM emaillists"); while (rs.next()) {

// You need to edit this column name String s = rs.getString("Name"); String s1 = rs.getString("email"); String s2 = rs.getString("website");

%>

<tr> <td><%= s %></td>

<td><%= s1 %></td>

<td><%= s2 %></td></tr>

<% } %>

</table>

<% rs.close(); rs = null; stmt.close(); stmt = null;

conn.close(); // Return to connection pool

conn = null; // Make sure we do not close it twice

} catch (SQLException e) { out.print("Throw e" + e); } finally {

// Always make sure result sets and statements are closed, & the connection is returned to the pool

if (rs != null) {

try { rs.close(); } catch (SQLException e) { } rs = null;

}

if (stmt != null) {

try { stmt.close(); } catch (SQLException e) { } stmt = null;

}

if (conn != null) {

try { conn.close(); } catch (SQLException e) { } conn = null;

}

} %>

</body>

</html>

# Programs:

**Consider the same program of practical 7**

Create an HTML form to calculate the incentive for the employee based on performance along with validation

* + A text box to accept employee name
  + A text box to accept employee salary
  + A radio group of 2 representing the designation (Manager and Programmer )
  + A drop down for Performance( Excellent, Good and Average)
  + A submit button named Calculate

When the calculate button is clicked, it should be redirected to JSP page which will calculate the incentive based on following criteria. If Performance is excellent then incentive is 15% of salary, if performance is good then incentive is 10% of salary and if performance is average then incentive is 5% of salary. If Designation is Manager then employee will get 3% more incentive.

Create a database named Company and create the table named employee. Insert all the details of the employee along with calculated incentive and total salary into the table. Display the message record inserted. This jsp page should be redirected to another jsp page which will fetch all the employee records and print it in tabular format.

# Part B (to be completed by students)

**(Students must submit the soft copy as per the following segments. Save your file as Rollno\_Name\_Practical8)**

|  |  |
| --- | --- |
| **Roll no.:** A001 | **Name:** Spoorthi Amin |
| **Class:** MCA | **Batch:** B1 |
| **Date of Experiment:** 13-09-2022 | **Date of Submission:** 14-09-2022 |

* 1. **Program scenario and Program code: (**Paste you program code )

**index.html**

<!DOCTYPE html>

<!--

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

-->

<html>

<head>

<title>Form</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

\*{

font-family: Verdana;

}

.container{

width: fit-content;

padding: 50px 50px 50px 50px;

margin-left: auto;

margin-right: auto;

background-color: whitesmoke;

}

</style>

</head>

<body style="background-color: aqua">

<div class="container">

<form action="message.jsp" method="post">

<h3>Calculate Salary</h3>

<hr>

Employee Name:<br>

<input type="text" name="ename" value="" required/>

<br><br>

Employee Salary:<br>

<input type="text" name="esal" value="" required/>

<br><br>

Designation:<br>

<input type="radio" name="r1" value="Manager" />Manager<br>

<input type="radio" name="r1" value="Programmer" />Programmer

<br><br>

Performance:<br>

<select name="performance">

<option>Excellent</option>

<option>Good</option>

<option>Average</option>

</select>

<br><br>

<input type="submit" value="CALCULATE" name="cal" style="width:100px;"/>

<input type="reset" value="RESET" style="margin-left: 20px; width: 100px"/>

</form>

</div>

</body>

</html>

**message.jsp**

<%--

Document : result

Created on : 8 Sep, 2022, 7:56:32 PM

Author : friyana\_\_000

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.Connection"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<h3>Employee Details</h3>

<%

String name= request.getParameter("ename");

int salary= Integer.parseInt(request.getParameter("esal"));

String designation= (String)request.getParameter("r1");

String performance= (String)request.getParameter("performance");

int incentive;

int total\_sal;

if(performance.equals("Excellent"))

{

incentive= (15\*salary)/100;

if(designation.equals("Manager"))

{

incentive= incentive+ ((3\*salary)/100);

total\_sal= salary+incentive;

}

else

{

total\_sal= salary+incentive;

}

}

else if(performance.equals("Good"))

{

incentive= (10\*salary)/100;

if(designation.equals("Manager"))

{

incentive= incentive+ ((3\*salary)/100);

total\_sal= salary+incentive;

}

else

{

total\_sal= salary+incentive;

}

}

else

{

incentive= (5\*salary)/100;

if(designation.equals("Manager"))

{

incentive= incentive+ ((3\*salary)/100);

total\_sal= salary+incentive;

}

else

{

total\_sal= salary+incentive;

}

}

out.println("Name: " + name + "<br>");

out.println("Designation: " + designation + "<br>");

out.println("Performance: " + performance + "<br>");

out.println("Salary: " + salary + "<br>");

out.println("Incentive: " + incentive + "<br>");

out.println("Total Salary: " + total\_sal + "<br>");

%>

<br>

<br>

<%

Class.forName("org.apache.derby.jdbc.ClientDriver") ;

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/testdb","root","root");

Statement stmt = con.createStatement();

int i= stmt.executeUpdate("insert into EMPLOYEE\_DETAILS\_TABLE(ename, designation, performance, salary, incentive, total\_sal) values ('"+name+"','"+designation+"','"+performance+"','"+salary+"','"+incentive+"','"+total\_sal+"')");

out.println("<h3>Employee Record Inserted Successfully!</h3>");

%>

<br>

<br>

<form action="result.jsp">

<h3>Click button to view Employee Database: </h3>

<input type="submit" value="DISPLAY">

</form>

</body>

</html>

**result.jsp**

<%--

Document : result

Created on : 8 Sep, 2022, 7:56:32 PM

Author : friyana\_\_000

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.Connection"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

<style>

table, th, td{

border: 1px solid black;

border-collapse: collapse;

padding-left: 10px;

padding-right: 10px;

}

.container{

width:fit-content;

margin-left: auto;

margin-right: auto;

}

</style>

</head>

<body>

<div class="container">

<h3 style="text-align: center">Employee Details</h3>

<hr>

<br><br>

<table>

<tr>

<th>Employee Name</th>

<th>Designation</th>

<th>Performance</th>

<th>Salary</th>

<th>Incentive</th>

<th>Total Salary</th>

</tr>

<%

Class.forName("org.apache.derby.jdbc.ClientDriver") ;

Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/testdb","root","root");

Statement stmt = con.createStatement();

String str = "select \* from EMPLOYEE\_DETAILS\_TABLE" ;

ResultSet rs= stmt.executeQuery(str);

while(rs.next()){

%>

<tr>

<td><%=rs.getString("ename") %> </td>

<td><%=rs.getString("designation") %> </td>

<td><%=rs.getString("performance") %> </td>

<td><%=rs.getString("salary") %> </td>

<td><%=rs.getString("incentive") %> </td>

<td><%=rs.getString("total\_sal") %> </td>

</tr>

<%

}

%>

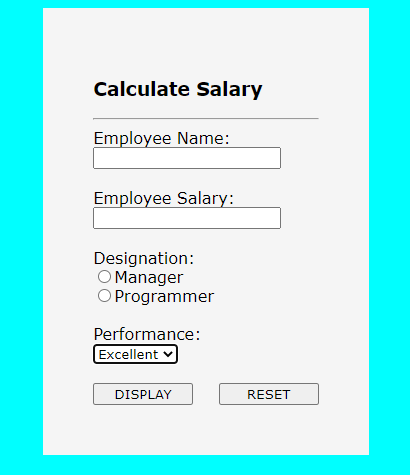
</table>

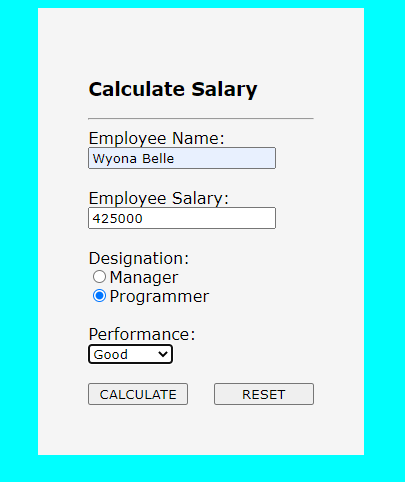
</div>

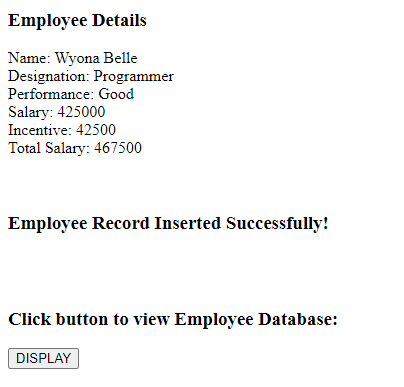
</body>

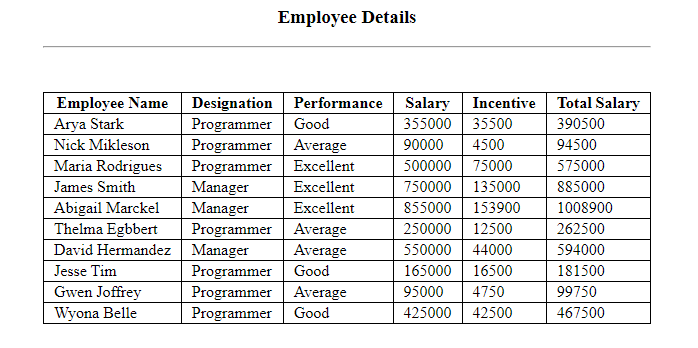
</html>

* 1. **Output: (**Paste Output of all web pages)



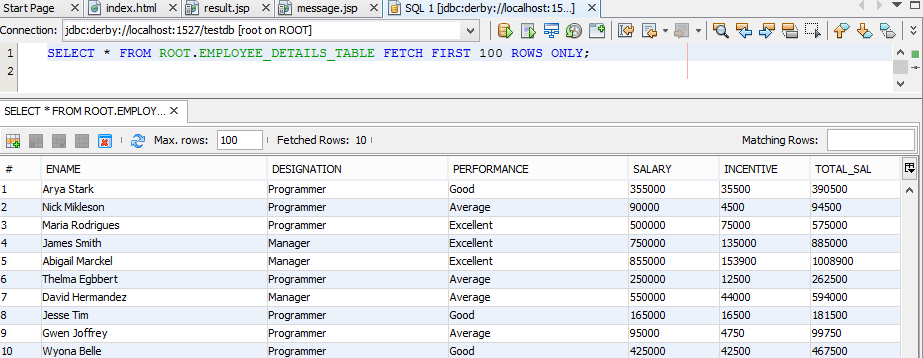
****

****

****

Graphical user interface, text, application

Description automatically generated



* 1. **Observation learning and conclusion:** mention what learning you got out of practical

Java Server Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

# Questions:

* + 1. **The syntax to include any package in JSP is <%@ import package**

**="java.sql.\*" %> (True/False). Justify.**

False.

To include any package in JSP the syntax is:-

<%@ page import = "java.sql.\*" %>

* + 1. **The** Statement **is simple SQL statement and takes no parameters, execute and compile every time when request is generated to database server.**
    2. **The** ResultSet **is an object which is used to store the result of executed SQL statement.**
    3. **What is difference between Statement and PreparedStatement?**

|  |  |
| --- | --- |
| Statement | PreparedStatement |
| It is used when SQL query is to be executed only once. | It is used when SQL query is to be executed multiple times. |
| You cannot pass parameters at runtime. | You can pass parameters at runtime. |
| Used for CREATE, ALTER, DROP statements. | Used for the queries which are to be executed multiple times. |
| Performance is very low. | Performance is better than Statement. |
| It is base interface. | It extends statement interface. |
| Used to execute normal SQL queries. | Used to execute dynamic SQL queries. |
| We cannot use statement for reading binary data. | We can use PreparedStatement for reading binary data. |
| It is used for DDL statements. | It is used for any SQL Query. |
| We cannot use statement for writing binary data. | We can use PreparedStatement for writing binary data. |
| No binary protocol is used for communication. | Binary protocol is used for communication. |