Written Exam

**EHSDI Unit: EH203 (Enterprise Java)**

**Time: 2 hours**

**Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Take time to read the questions carefully and write your answers clearly on this paper.

**1. Describe the different components of the *Model View Controller* pattern and how they interact with each other.**

***3 marks***

**2. Below is a snippet from the web.xml file of a web application called *testemr*, running on the *localhost* server. It contains 3 servlets and one html page in *WEB-INF* called *index.html*. Complete the table stating what is returned when each of the given URLs are requested, i.e. which servlet or which page.**

***3 marks***

<servlet-mapping>

<servlet-name>TestServlet1</servlet-name>

<url-pattern>/home</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>TestServlet2</servlet-name>

<url-pattern>/testing/\*</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>TestServlet3</servlet-name>

<url-pattern>\*.jsp</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

</welcome-file-list>

|  |  |
| --- | --- |
| **Requested URL** | **Returns** |
| http://localhost/testemr/ |  |
| http://localhost/testemr/index.html |  |
| http://localhost/testemr/home |  |
| http://localhost/testemr/home.jsp |  |
| http://localhost/testemr/testing/index.html |  |

**3 (a). Give an example of when it is good to store website data in the session.**

***3 marks***

**3 (b). Every session managed by a J2EE web server has a unique session ID. Describe how this ID value is associated with a specific client.**

***3 marks***

**4. Complete the following table which describes the different elements that can be used in a Java Server Page:**

***6 marks***

|  |  |  |
| --- | --- | --- |
| **Name** | **Example** | **Description** |
| *Scriptlet* | *<% n = 5; %>* | *Puts Java code directly into the servlet service method* |
|  | <%= "Hello" %> |  |
| Declaration |  |  |
|  | <%@ page %> |  |
| Comment |  |  |

**5. Given the following code from a servlet’s doGet method, complete the table below which shows how the values can be accessed using Expression Language in a JSP:**

***5 marks***

request.setAttribute("age", 55);

request.setAttribute("name", "Bob Smith");

request.setAttribute("ids", new int[] {73, 53, 81});

|  |  |
| --- | --- |
| **Java value in servlet** | **Expression Language in JSP** |
| age | ${age} |
| name |  |
| ids[1] |  |
| name.getBytes() |  |
| request.getParameter("id") |  |
| request.getHeader("Referer") |  |

**7 (a). Given the following code in a Java Server Page, what will the HTML output be?**

***3 marks***

<%

String[] vals = { "Hello", "<br/>", null, "world", "<i>now</i>"};

request.setAttribute("vals", vals);

%>

<c:forEach items="${vals}" var="val">

<c:choose>

<c:when test="${fn:length(val) > 5}">

<c:out value="${val}" />

</c:when>

<c:otherwise>

${val ne null ? val : 'X'}

</c:otherwise>

</c:forEach>

**8. What are the two different ways of creating a custom tag in a web application?**

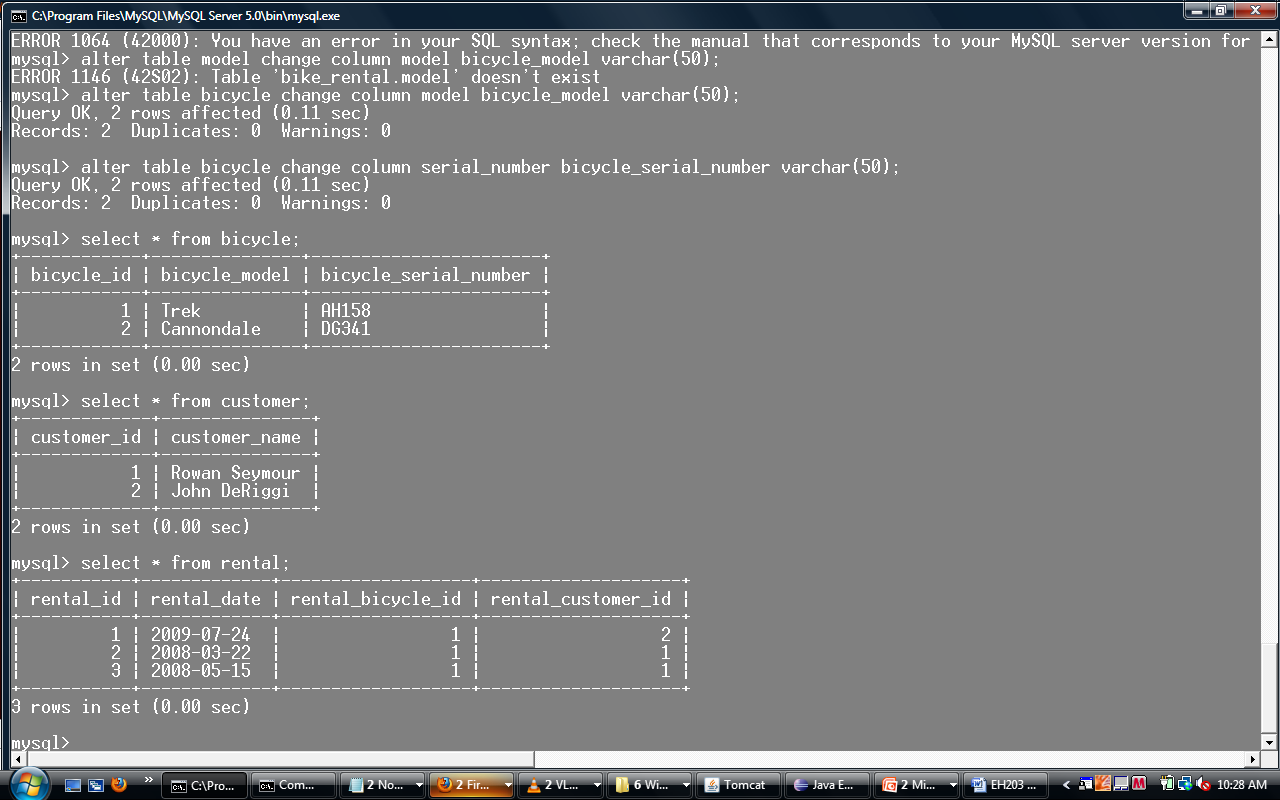
**8. What are the 3 properties specifications of a tag attribute in custom tag? Explain the properties.**

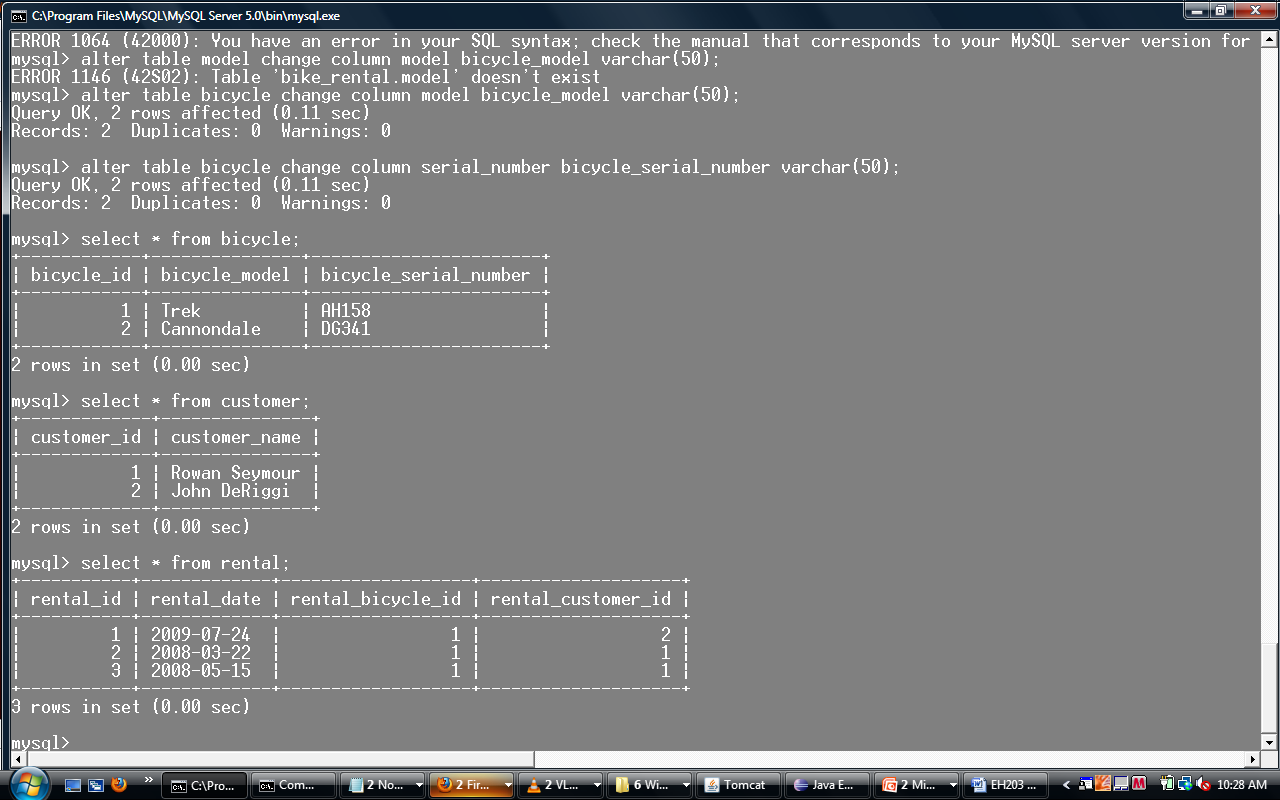
***2 marks***

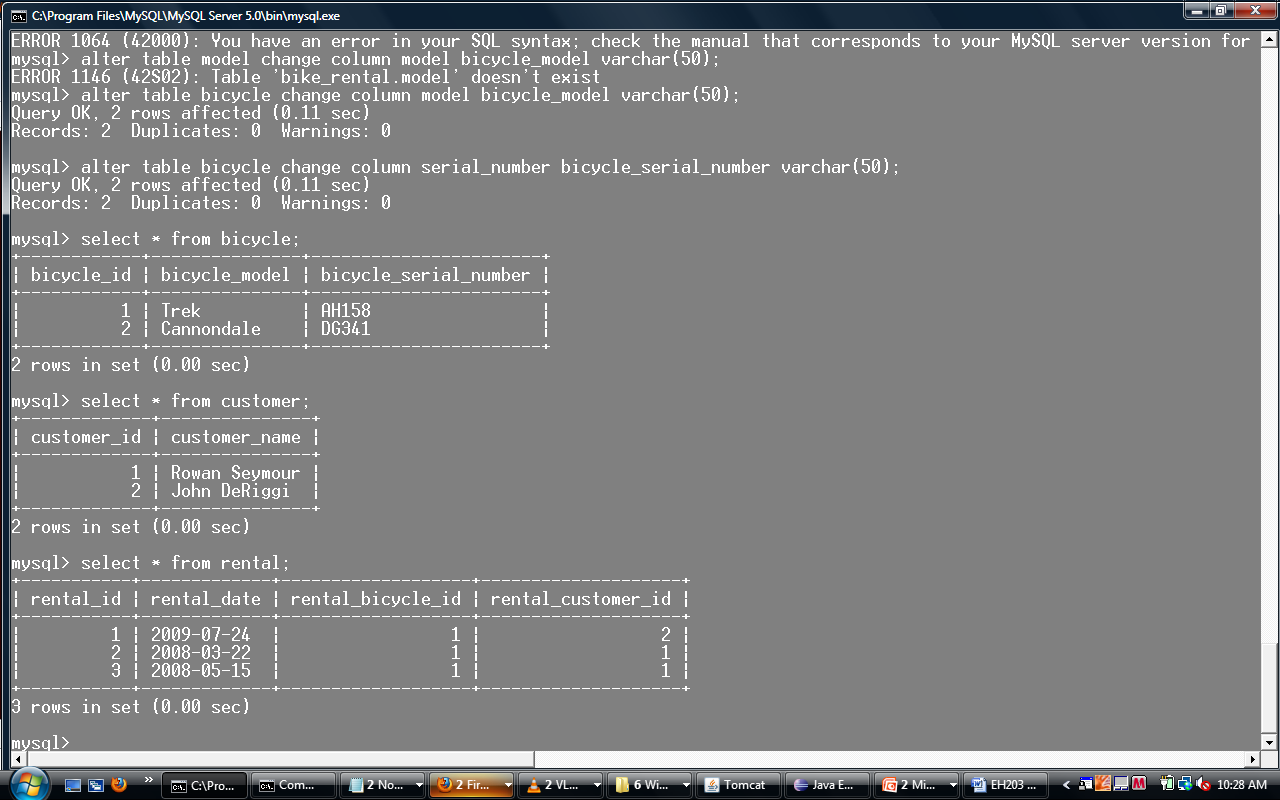
**9. Describe what a *Filter* is and give an example of how one might be used in a J2EE web application.**

***3 marks***

**10. There is a database called *bicycle\_rental* which is used to keep track of bicycles and the customers to whom they are rented. The database consists of the following tables: *bicycle*, *customer* and *rental*.**

****

****

****

**(a) Write a join query to show the names of the customers, the date on which the customer rented the bicycle and the model of the bicycle that was rented.**

***3 marks***

**(b) Write a query for the number of times each *rental\_customer\_id* rented a bicycle.**

***3 marks***

**11. (a) Given the following Hibernate XML mapping, draw an entity relationship diagram for the *House* and *Resident* entities:**

***3 marks***

**<hibernate-mapping package="ehsdi.domain">**

**<class name="Resident" table="residents">**

**<id name="residentId" column="resident\_id">**

**<generator class="native" />**

**</id>**

**<property name="name" type="string" column="name" />**

**<many-to-one name="house" class="House" column="house\_id" />**

**</class>**

**</hibernate-mapping>**

**11. (b) Write suitable Java classes for the entities in your ER diagram.**

***3 marks***

**THE END OF THE TEST**