

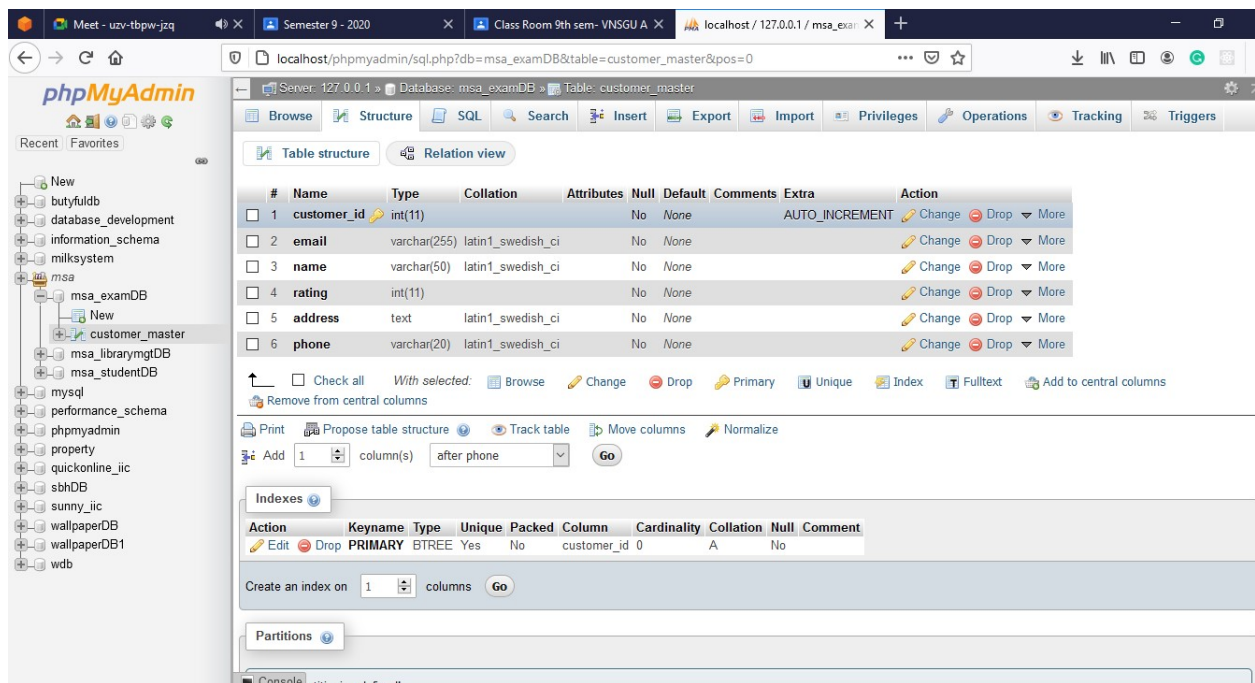
Name: Sanket H. Vanani

No: 30

Sem: IT 9th

Subject : Cloud Computing

1. Create Database in mysql through PHPMysqladmin.



The screenshot displays the PHPMysqladmin web interface. The left sidebar shows a tree view of databases, with 'msa_examDB' selected. The main panel shows the 'Table structure' view for the 'customer_master' table. The table has six columns: 'customer_id' (int(11), PRIMARY, AUTO_INCREMENT), 'email' (varchar(255)), 'name' (varchar(50)), 'rating' (int(11)), 'address' (text), and 'phone' (varchar(20)). The 'Indexes' section shows a PRIMARY index on 'customer_id'. The 'Partitions' section is empty.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	customer_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	rating	int(11)			No	None			Change Drop More
5	address	text	latin1_swedish_ci		No	None			Change Drop More
6	phone	varchar(20)	latin1_swedish_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	customer_id	0	A	No	

Create an index on 1 column(s) after phone [Go](#)

Partitions

2. Add 2-3 data for testing.

Server: 127.0.0.1 » Database: msa_examDB » Table: customer_master

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking

Showing rows 0 - 1 (2 total, Query took 0.0030 seconds.)

SELECT * FROM `customer_master`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [C

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	customer_id	email	name	rating	address	phone
<input type="checkbox"/> Edit Copy Delete	1	sanket@gmail.com	sanket	5	surat	7643312448
<input type="checkbox"/> Edit Copy Delete	2	sanket2@gmail.com	sanket2	7	vadodara	54621365

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

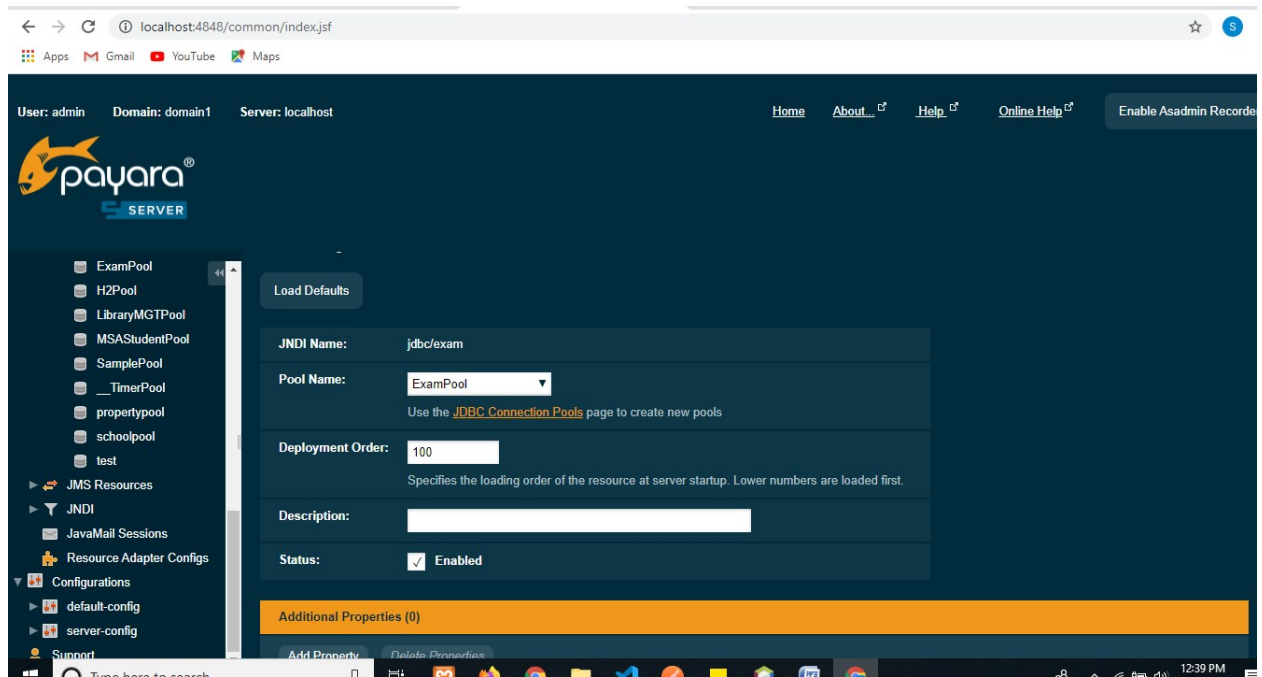
Query results operations

Print Copy to clipboard Export Display chart Create view

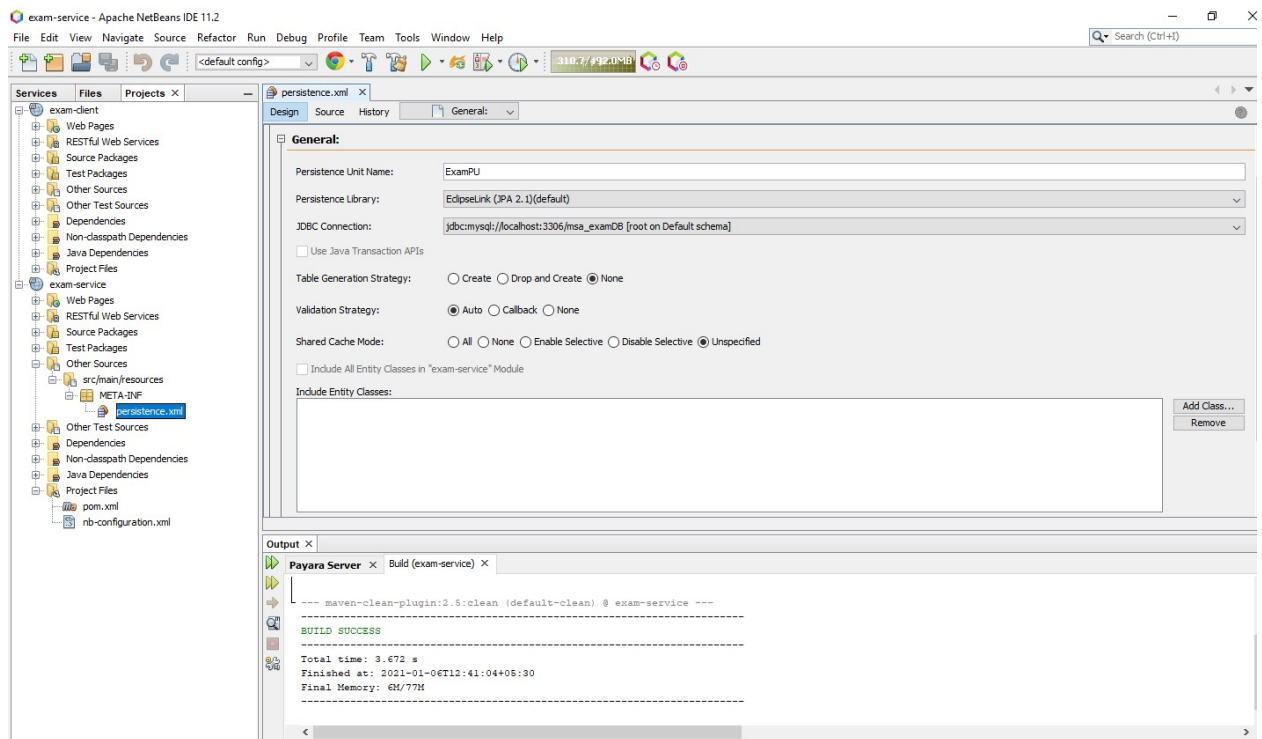
Bookmark this SQL query

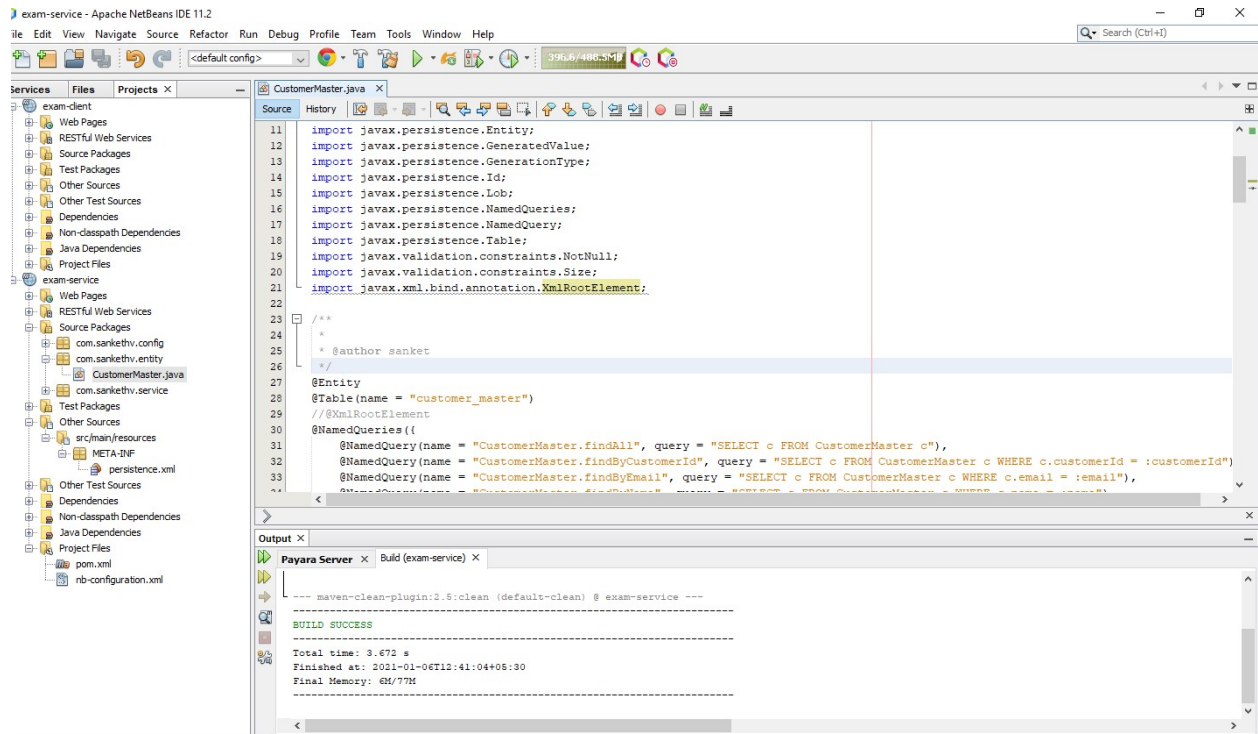
3. For creating MSA Application we need to make Project with Archtype and I have created 2 project one called exam-client and another one is exam-service.

4. Then we need to create the JDBC pool and JDBC resource in Payara Server.

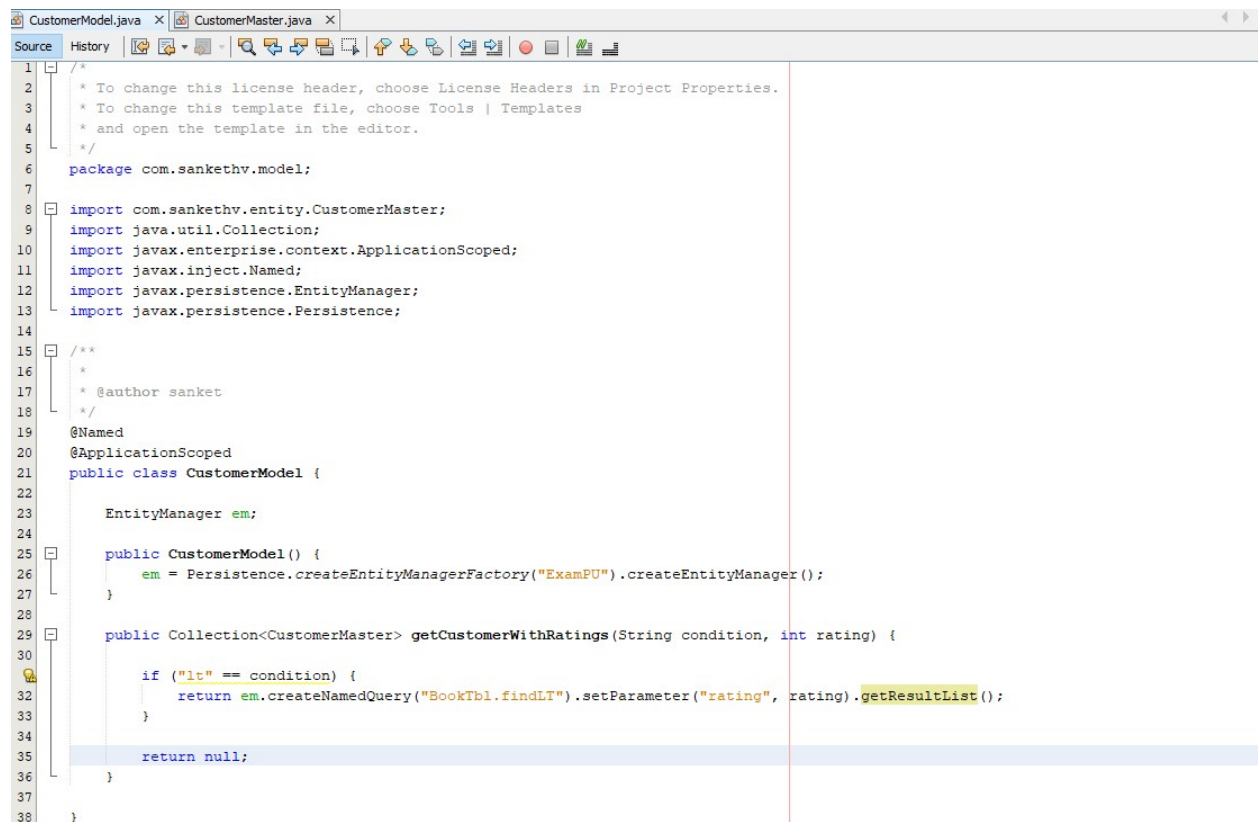


5. Now we will create Persistence unit and Entity from the database.





6. Now we will create the model Customer and add query as per our requirements in Entity.



```
/*
```

```
 * 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.
```

```
*/
```

```
package com.sankethv.model;
```

```
import com.sankethv.entity.CustomerMaster;
```

```
import java.util.Collection;
```

```
import javax.enterprise.context.ApplicationScoped;
```

```
import javax.inject.Named;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.Persistence;
```

```
/**
```

```
 *
```

```
 * @author sanket
```

```
 */
```

```
@Named
```

```
@ApplicationScoped
```

```
public class CustomerModel {
```

```
    EntityManager em;
```

```
    public CustomerModel() {
```

```

        em = Persistence.createEntityManagerFactory("ExamPU").createEntityManager();
    }

    public Collection<CustomerMaster> getCustomerWithRatings(String condition, int rating) {

        if ("lt" == condition) {

            return em.createNamedQuery("BookTbl.findLT").setParameter("rating", rating).getResultList();

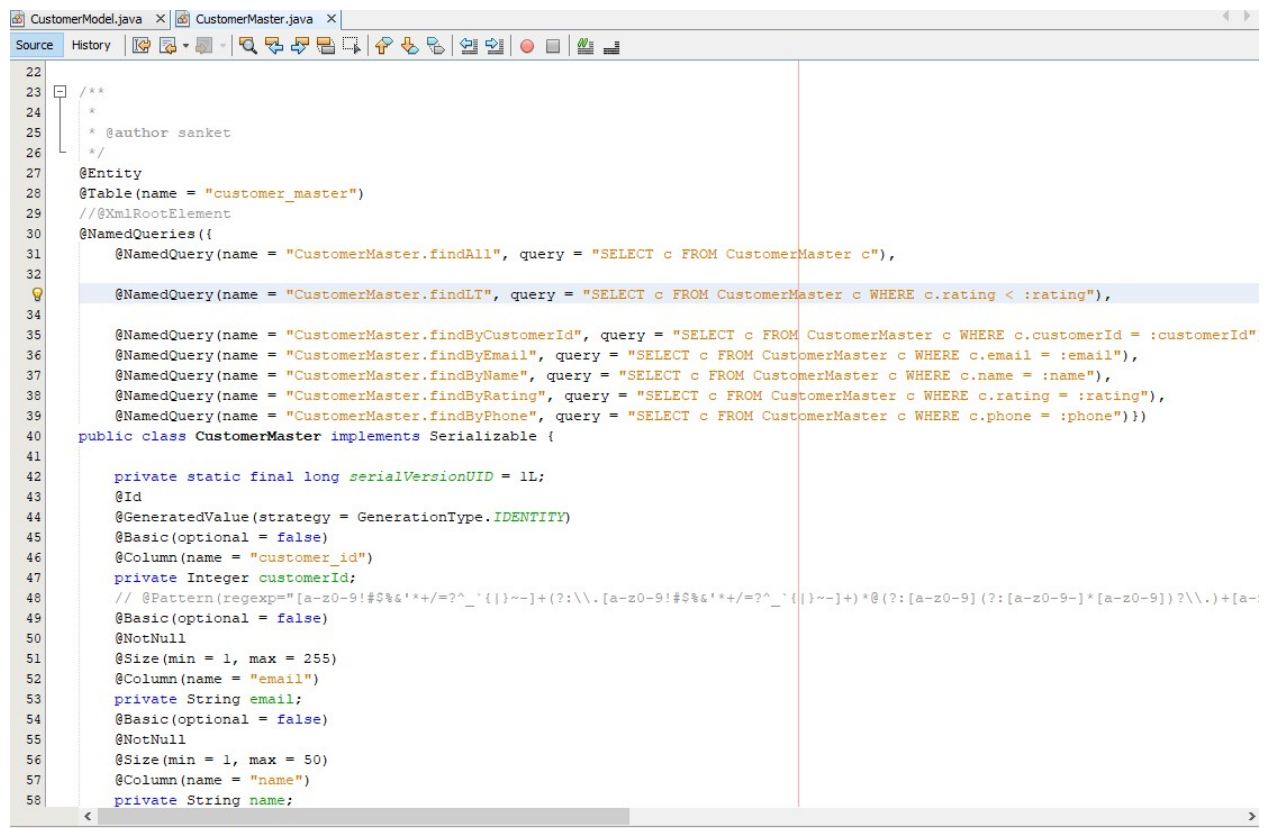
        }

        return null;

    }

}

```



```

22
23  /**
24   *
25   * @author sanket
26   */
27  @Entity
28  @Table(name = "customer_master")
29  @XmlRootElement
30  @NamedQueries({
31      @NamedQuery(name = "CustomerMaster.findAll", query = "SELECT c FROM CustomerMaster c"),
32      @NamedQuery(name = "CustomerMaster.findLT", query = "SELECT c FROM CustomerMaster c WHERE c.rating < :rating"),
33
34      @NamedQuery(name = "CustomerMaster.findById", query = "SELECT c FROM CustomerMaster c WHERE c.customerId = :customerId"),
35      @NamedQuery(name = "CustomerMaster.findByEmail", query = "SELECT c FROM CustomerMaster c WHERE c.email = :email"),
36      @NamedQuery(name = "CustomerMaster.findByName", query = "SELECT c FROM CustomerMaster c WHERE c.name = :name"),
37      @NamedQuery(name = "CustomerMaster.findByRating", query = "SELECT c FROM CustomerMaster c WHERE c.rating = :rating"),
38      @NamedQuery(name = "CustomerMaster.findByPhone", query = "SELECT c FROM CustomerMaster c WHERE c.phone = :phone"))
39
40  public class CustomerMaster implements Serializable {
41
42      private static final long serialVersionUID = 1L;
43      @Id
44      @GeneratedValue(strategy = GenerationType.IDENTITY)
45      @Basic(optional = false)
46      @Column(name = "customer_id")
47      private Integer customerId;
48      // @Pattern(regexp="[a-z0-9!#$%&'*/=?^_`{|}~]+(?:\\.[a-z0-9!#$%&'*/=?^_`{|}~]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.\\.)+[a-z0-9!#$%&'*/=?^_`{|}~]+(?:\\.[a-z0-9!#$%&'*/=?^_`{|}~]+)*"
49      @Basic(optional = false)
50      @NotNull
51      @Size(min = 1, max = 255)
52      @Column(name = "email")
53      private String email;
54      @Basic(optional = false)
55      @NotNull
56      @Size(min = 1, max = 50)
57      @Column(name = "name")
58      private String name;

```

```
/*
```

```
 * 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.
```

```
 */
```

```
package com.sankethv.entity;
```

```
import java.io.Serializable;
```

```
import javax.persistence.Basic;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Lob;
```

```
import javax.persistence.NamedQueries;
```

```
import javax.persistence.NamedQuery;
```

```
import javax.persistence.Table;
```

```
import javax.validation.constraints.NotNull;
```

```
import javax.validation.constraints.Size;
```

```
import javax.xml.bind.annotation.XmlRootElement;
```

```
/**
```

```
 *
```

```
 * @author sanket
```

***/**

@Entity

@Table(name = "customer_master")

//@XmlElement

@NamedQueries({

@NamedQuery(name = "CustomerMaster.findAll", query = "SELECT c FROM CustomerMaster c"),

**@NamedQuery(name = "CustomerMaster.findLT", query = "SELECT c FROM CustomerMaster c
WHERE c.rating < :rating"),**

**@NamedQuery(name = "CustomerMaster.findByCustomerId", query = "SELECT c FROM
CustomerMaster c WHERE c.customerId = :customerId"),**

**@NamedQuery(name = "CustomerMaster.findByEmail", query = "SELECT c FROM CustomerMaster
c WHERE c.email = :email"),**

**@NamedQuery(name = "CustomerMaster.findByName", query = "SELECT c FROM CustomerMaster
c WHERE c.name = :name"),**

**@NamedQuery(name = "CustomerMaster.findByRating", query = "SELECT c FROM CustomerMaster
c WHERE c.rating = :rating"),**

**@NamedQuery(name = "CustomerMaster.findByPhone", query = "SELECT c FROM CustomerMaster
c WHERE c.phone = :phone"))}**

public class CustomerMaster implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Basic(optional = false)

@Column(name = "customer_id")

private Integer customerId;

// @Pattern(regexp="[a-z0-9!#\$%&'*/+=?^_`{|}~-]+(?:\\.[a-z0-9!#\$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9]*[a-z0-9])?", message="Invalid email")//if the field contains email address consider using this annotation to enforce field validation

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 255)

@Column(name = "email")

private String email;

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 50)

@Column(name = "name")

private String name;

@Basic(optional = false)

@NotNull

@Column(name = "rating")

private int rating;

@Basic(optional = false)

@NotNull

@Lob

@Size(min = 1, max = 65535)

@Column(name = "address")

private String address;

// @Pattern(regexp="^(?!(\\d{3})\\|)?[-]?(\\d{3})[-]?(\\d{4})\$", message="Invalid phone/fax format, should be as xxx-xxx-xxxx")//if the field contains phone or fax number consider using this annotation to enforce field validation

@Basic(optional = false)

@NotNull

@Size(min = 1, max = 20)

@Column(name = "phone")

private String phone;

public CustomerMaster() {

}

public CustomerMaster(Integer customerId) {

this.customerId = customerId;

}

**public CustomerMaster(Integer customerId, String email, String name, int rating, String address,
String phone) {**

this.customerId = customerId;

this.email = email;

this.name = name;

this.rating = rating;

this.address = address;

this.phone = phone;

}

public Integer getCustomerId() {

return customerId;

}

```
public void setCustomerId(Integer customerId) {  
    this.customerId = customerId;  
}
```

```
public String getEmail() {  
    return email;  
}
```

```
public void setEmail(String email) {  
    this.email = email;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public int getRating() {  
    return rating;  
}
```

```
public void setRating(int rating) {
```

```
    this.rating = rating;
}
```

```
public String getAddress() {
    return address;
}
```

```
public void setAddress(String address) {
    this.address = address;
}
```

```
public String getPhone() {
    return phone;
}
```

```
public void setPhone(String phone) {
    this.phone = phone;
}
```

```
@Override
```

```
public int hashCode() {
    int hash = 0;
    hash += (customerId != null ? customerId.hashCode() : 0);
    return hash;
}
```

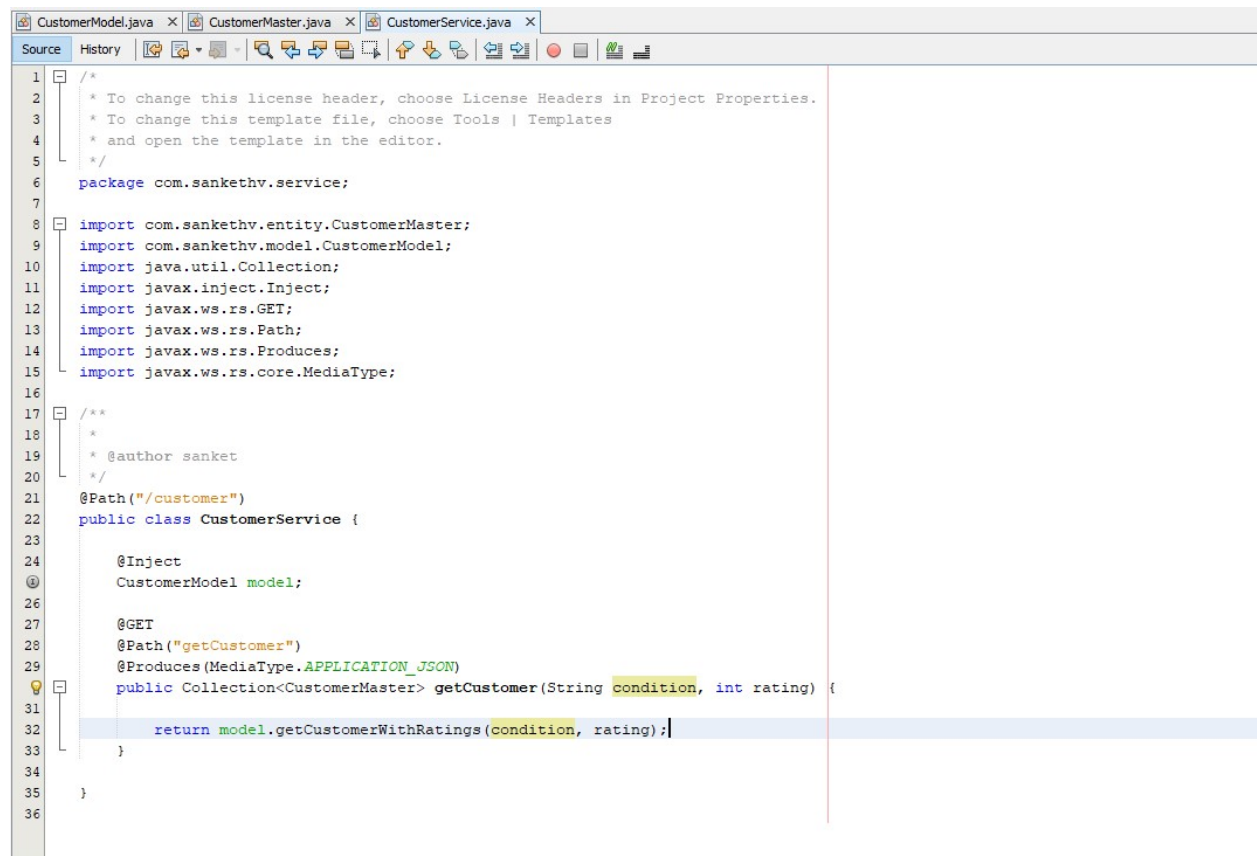
@Override

```
public boolean equals(Object object) {  
  
    // TODO: Warning - this method won't work in the case the id fields are not set  
  
    if (!(object instanceof CustomerMaster)) {  
  
        return false;  
  
    }  
  
    CustomerMaster other = (CustomerMaster) object;  
  
    if ((this.customerId == null && other.customerId != null) || (this.customerId != null &&  
!this.customerId.equals(other.customerId))) {  
  
        return false;  
  
    }  
  
    return true;  
  
}
```

@Override

```
public String toString() {  
  
    return "com.sankethv.entity.CustomerMaster[ customerId=" + customerId + " ]";  
  
}  
  
}
```

7. Now create the service for testing.



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package com.sankethv.service;
7
8  import com.sankethv.entity.CustomerMaster;
9  import com.sankethv.model.CustomerModel;
10 import java.util.Collection;
11 import javax.inject.Inject;
12 import javax.ws.rs.GET;
13 import javax.ws.rs.Path;
14 import javax.ws.rs.Produces;
15 import javax.ws.rs.core.MediaType;
16
17 /**
18 *
19 * @author sanket
20 */
21 @Path("/customer")
22 public class CustomerService {
23
24     @Inject
25     CustomerModel model;
26
27     @GET
28     @Path("getCustomer")
29     @Produces(MediaType.APPLICATION_JSON)
30     public Collection<CustomerMaster> getCustomer(String condition, int rating) {
31
32         return model.getCustomerWithRatings(condition, rating);
33     }
34
35 }
36
```

package com.sankethv.service;

import com.sankethv.entity.CustomerMaster;

import com.sankethv.model.CustomerModel;

import java.util.Collection;

import javax.inject.Inject;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

```
/**
 *
 * @author sanket
 */
@Path("/customer")
public class CustomerService {

    @Inject
    CustomerModel model;

    @POST
    @Path("getCustomer")
    @Produces(MediaType.APPLICATION_JSON)
    @Consumes(MediaType.APPLICATION_JSON)
    public Collection<CustomerMaster> getCustomer(String condition, int rating) {

        return model.getCustomerWithRatings(condition, rating);
    }

}
```