

application.properties

TableScript.sql

Movie.java

Mo

```
1 package com.infy.entity;
2
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.Id;
6
7 @Entity
8 public class Movie {
9     @Id
10     private Integer movieId;
11     private String movieName;
12     private String directorName;
13     private Float imdbRating;
14     @Column(name="release_year")
15     private Integer year;
16     public Integer getMovieId() {
17         return movieId;
18     }
19     public void setMovieId(Integer movieId) {
20         this.movieId = movieId;
21     }
22     public String getMovieName() {
23         return movieName;
24     }
25     public void setMovieName(String movieName) {
26         this.movieName = movieName;
27     }
28     public String getDirectorName() {
29         return directorName;
30     }
31     public void setDirectorName(String directorName) {
32         this.directorName = directorName;
33     }
34     public Float getImdbRating() {
```

Console

<terminated> Inf.PlexImplApplication [Java Application] C:\Program Files\JDK\bin\1.8\bin



pplication.pr... TableScript.sql Movie.java MovieRepositor... InfyPlexImplAp... MovieRe

```
package com.infy.repository;
```

```
import java.util.ArrayList;
```

```
@Repository(value="movieRepository")
```

```
public class MovieRepositoryImpl implements MovieRepository {
```

```
@PersistenceContext
```

```
private EntityManager entityManager;
```

```
@Override
```

```
public MovieDTO getMovieByName(String movieName) {
```

```
    MovieDTO m=null;
```

```
    Query query=entityManager.createQuery("Select c from Movie c where c.movieName=?1");
```

```
    query.setParameter(1, movieName);
```

```
    Movie m1=(Movie) query.getSingleResult();
```

```
    if(m1!=null)
```

```
    {
```

```
        m=new MovieDTO();
```

```
        m.setDirectorName(m1.getDirectorName());
```

```
        m.setImdbRating(m1.getImdbRating());
```

```
        m.setMovieId(m1.getMovieId());
```

```
        m.setMovieName(m1.getMovieName());
```

```
        m.setYear(m1.getYear());
```

```
    }
```

```
    return m;
```

```
}
```

```
@Override
```

```
public List<MovieDTO> getMoviesByImdbRating(Double fromRating, Double toRating) {
```

```
    List<MovieDTO> l=new ArrayList<>();
```

```
    String queryString = "select c from Movie c where c.imdbRating>=?1 and c.imdbRating<=?2";
```

```
    Query query = entityManager.createQuery(queryString);
```

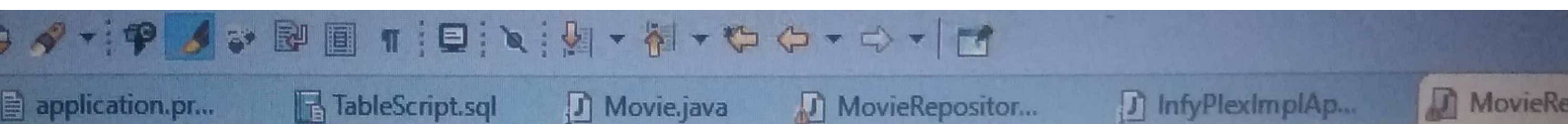
Problems Run Debug Console

minated> InfyPlexImplApplication [Java Application] C:\Program Files\Zulu\zulu-11\bin\javaw.exe (01-Mar-2021, 10:12:32 pm)

Writable

Smart Insert

1:1:0



```
44
45     Query query = entityManager.createQuery(queryString);
46     Float a=Float.valueOf(fromRating.toString());
47
48     Float b=Float.valueOf(toRating.toString());
49     query.setParameter(1, a);
50     query.setParameter(2, b);
51     List<Movie> l1=query.getResultList();
52     if(!l1.isEmpty())
53     {
54         for(Movie m1:l1)
55         {
56             MovieDTO m=new MovieDTO();
57             m.setDirectorName(m1.getDirectorName());
58             m.setImdbRating(m1.getImdbRating());
59             m.setMovieId(m1.getMovieId());
60             m.setMovieName(m1.getMovieName());
61             m.setYear(m1.getYear());
62             l.add(m);
63         }
64     }
65     return l;
66
67 }
68
69 @Override
70 public List<Object[]> getMoviesNameAndYear(String directorName) {
71     String queryString = "select c.movieName,c.year from Movie c where c.directorName=?1";
72     Query query = entityManager.createQuery(queryString);
73     query.setParameter(1, directorName);
74     List<Object[]> result = query.getResultList();
75     return result;
76 }
77
78
```

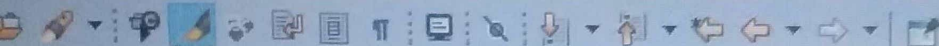
Problems in IDE: Declaration Console

<terminated> InfyPlexImplApplication [Java Application] C:\Program Files\Zulu\zulu-11\bin\javaw.exe (01-Mar-2021, 10:12:32 pm)

Writable

Smart Insert

1:1:0



application.pr... TableScript.sql Movie.java MovieRepositor... InfyPlexImplAp... MovieRepositor...

```
1 package com.infy.repository;
2
3 import java.util.ArrayList;
4
14
15 @Repository(value="movieRepository")
16 public class MovieRepositoryImpl implements MovieRepository {
17
18     @PersistenceContext
19     private EntityManager entityManager;
20
21     @Override
22     public List<MovieDTO> getMovieByRating(Double fromRating) {
23         List<MovieDTO> l=new ArrayList<>();
24         String queryString = "select c from Movie c where c.imdbRating>=?1 order by c.imdbRating";
25
26         Query query = entityManager.createQuery(queryString);
27         Float a=Float.valueOf(fromRating.toString());
28
29         query.setParameter(1, a);
30
31         List<Movie> l1=query.getResultList();
32         if(!l1.isEmpty())
33         {
34             for(Movie m1:l1)
35             {
36                 MovieDTO m=new MovieDTO();
37                 m.setDirectorName(m1.getDirectorName());
38                 m.setImdbRating(m1.getImdbRating());
39                 m.setMovieId(m1.getMovieId());
40                 m.setMovieName(m1.getMovieName());
41                 m.setYear(m1.getYear());
42                 l.add(m);
43             }
44         }
```

Console

<terminated> InfyPlexImplApplication [Java Application] C:\Program Files\Zulu\zulu-11\bin\javaw.exe (01-Mar-2021 10:12:32 pm)

Writable

Smart Insert

27 / 34 / 807

application.pr... TableScript.sql Movie.java *MovieRepositor... InfyPlexImplAp... MovieReposito

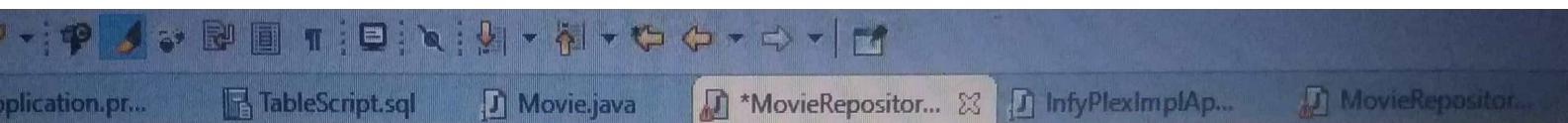
```
5   return l;  
6   }  
7  
8 @Override  
9 public List<MovieDTO> getHighestRatedMovie(String directorName) {  
0  
1   List<MovieDTO> l=new ArrayList<>();  
2   String queryString = "select c from Movie c where c.directorName=?1 and c.imdbRating=?2";  
3   Query query1 = entityManager.createQuery("select max(c.imdbRating) from Movie c."  
4       + "group by c.directorName having c.directorName=?1");  
5   Query query = entityManager.createQuery(queryString);  
6   query1.setParameter(1, directorName);  
7   Float i=(Float)query1.getSingleResult();  
8  
9   query.setParameter(1, directorName);  
0   query.setParameter(2, i);  
1   List<Movie> l1=query.getResultList();  
2   if(!l1.isEmpty())  
3   {  
4       for(Movie m1:l1)  
5       {  
6           MovieDTO m=new MovieDTO();  
7           m.setDirectorName(m1.getDirectorName());  
8           m.setImdbRating(m1.getImdbRating());  
9           m.setMovieId(m1.getMovieId());  
0           m.setMovieName(m1.getMovieName());  
1           m.setYear(m1.getYear());  
2           l.add(m);  
3       }  
4   }  
5   return l;  
6   }  
7  
8 @Override  
9 public Float getAverageDirectorRating(String directorName) {
```

Console
<terminated> InfyPlexImplApplication [Java Application] C:\Program Files\Zulu\zulu-11\bin\javaw.exe (01-Mar-2021, 10:12:32 pm)

Writable

Smart Insert

54/20/1517



```
MovieDTO m=new MovieDTO();
m.setDirectorName(m1.getDirectorName());
m.setImdbRating(m1.getImdbRating());
m.setMovieId(m1.getMovieId());
m.setMovieName(m1.getMovieName());
m.setYear(m1.getYear());
l.add(m);
}
}
return l;
}

@Override
public Float getAverageDirectorRating(String directorName) {
    Query query1 = entityManager.createQuery("select max(c.imdbRating) from Movie c "
        + "group by c.directorName having c.directorName=?1");
    query1.setParameter(1, directorName);
    Float i=(Float)query1.getSingleResult();
    return i;
}

@Override
public Long getNumberOfMoviesReleased(Integer fromYear, Integer toYear) {
    Query query1 = entityManager.createQuery("select count(c) from Movie c where c.year>?1 and c.year<?2");
    query1.setParameter(1, fromYear);
    query1.setParameter(2, toYear);
    Long i=(Long)query1.getSingleResult();
    return i;
}
}
```

Problems | JavaDoc | Declaration | Console

minated> InfyPlexImplApplication [Java Application] C:\Program Files\Zulu\zulu-11\bin\javaw.exe (01-Mar-2021, 10:12:32 pm)

Writable

Smart Insert

81 : 20 : 2317