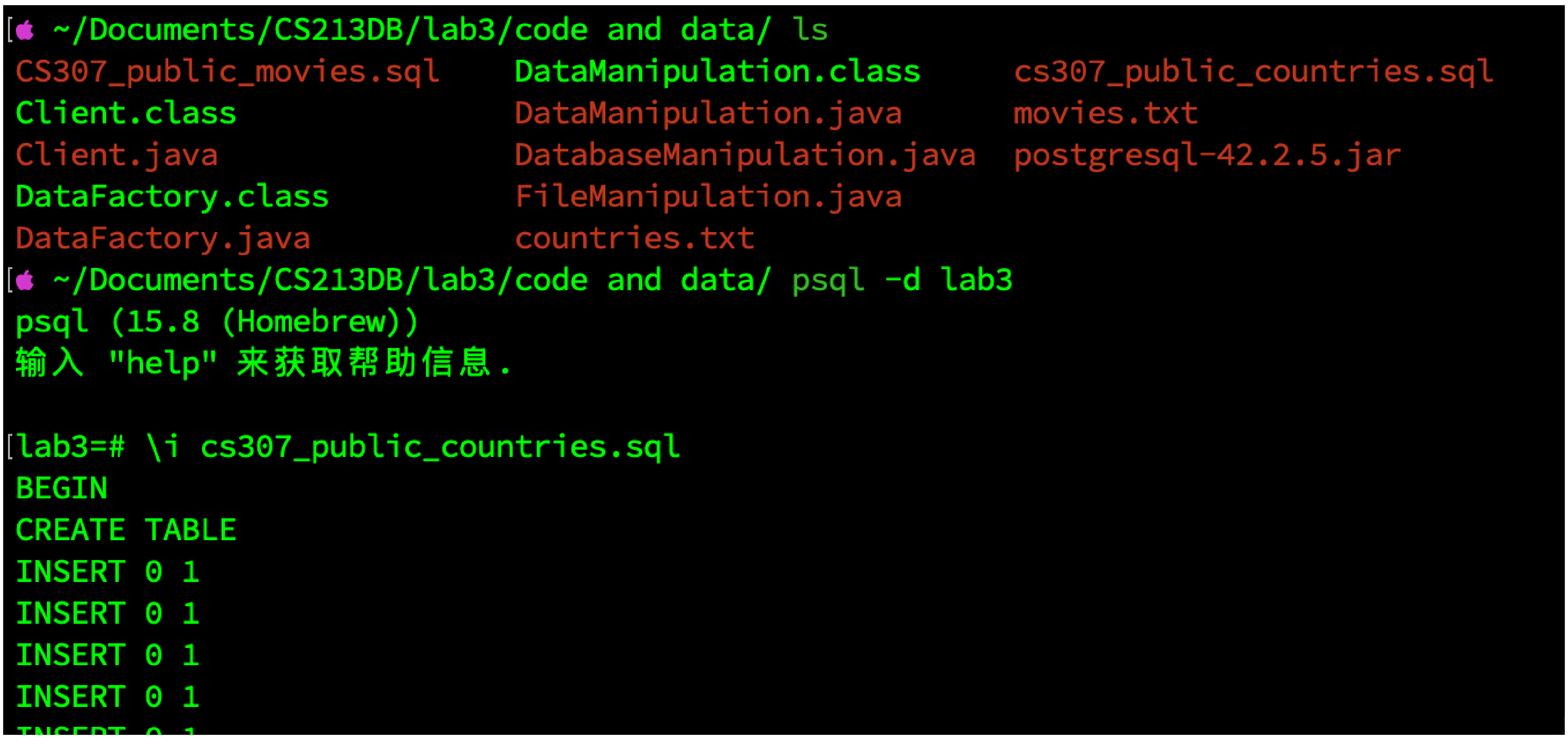
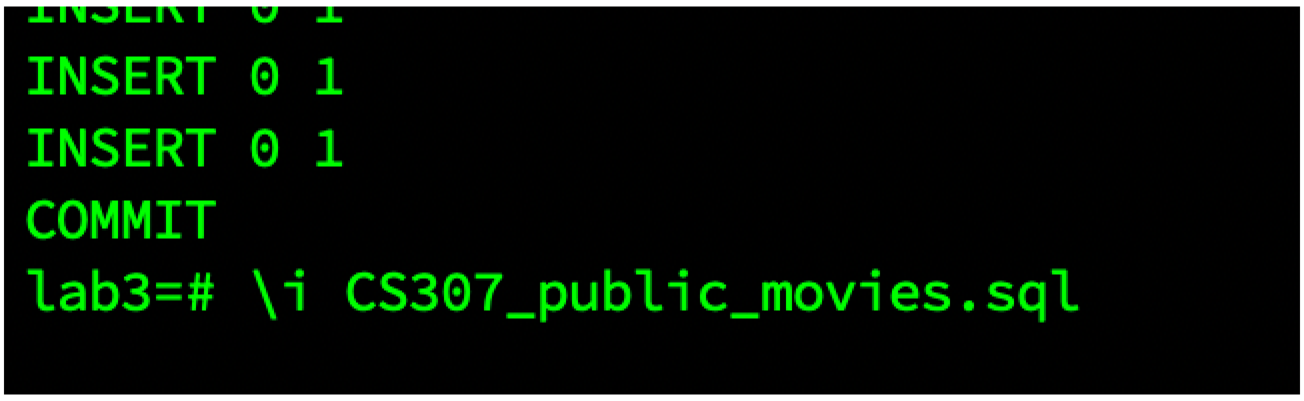
Lab3 record

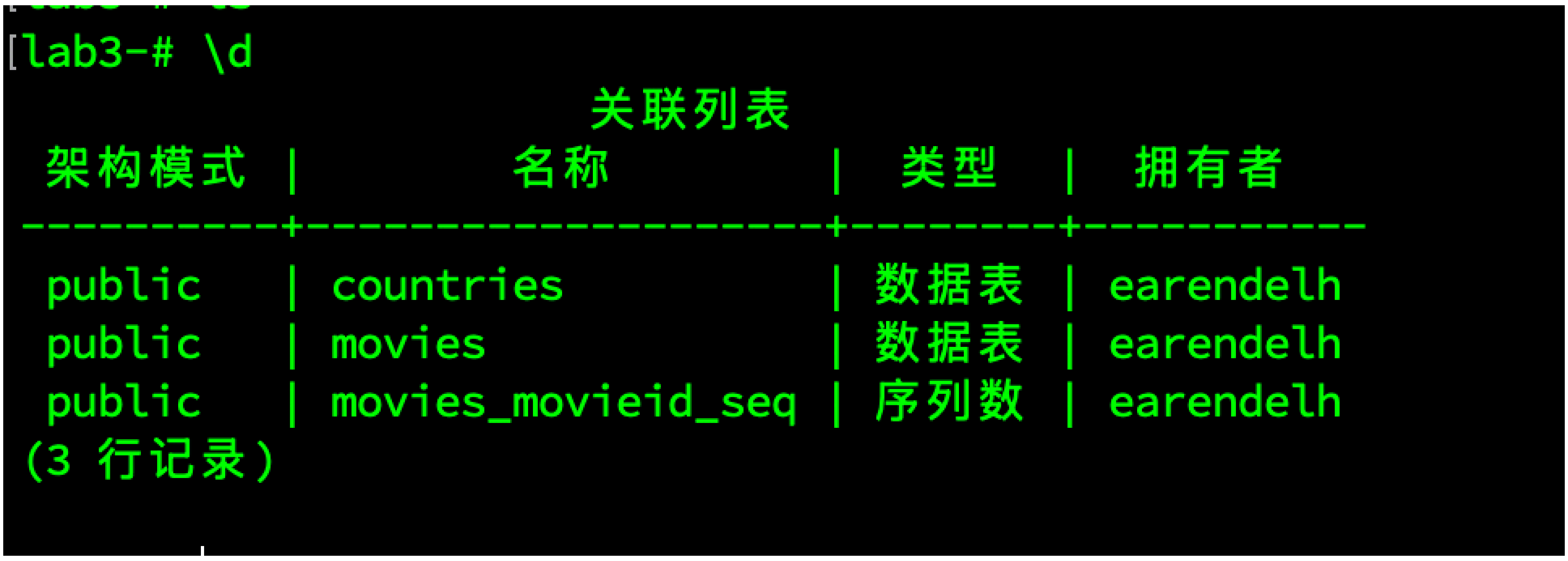
SID: 12310401
name: 王子恒



```
[ ~/Documents/CS213DB/lab3/code and data/ ls
CS307_public_movies.sql     DataManipulation.class     cs307_public_countries.sql
Client.class                DataManipulation.java      movies.txt
Client.java                 DatabaseManipulation.java  postgresql-42.2.5.jar
DataFactory.class           FileManipulation.java
DataFactory.java            countries.txt
[ ~/Documents/CS213DB/lab3/code and data/ psql -d lab3
psql (15.8 (Homebrew))
输入 "help" 来获取帮助信息.

[lab3=# \i cs307_public_countries.sql
BEGIN
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```



```
INSERT 0 1
INSERT 0 1
INSERT 0 1
COMMIT
lab3=# \i CS307_public_movies.sql
```

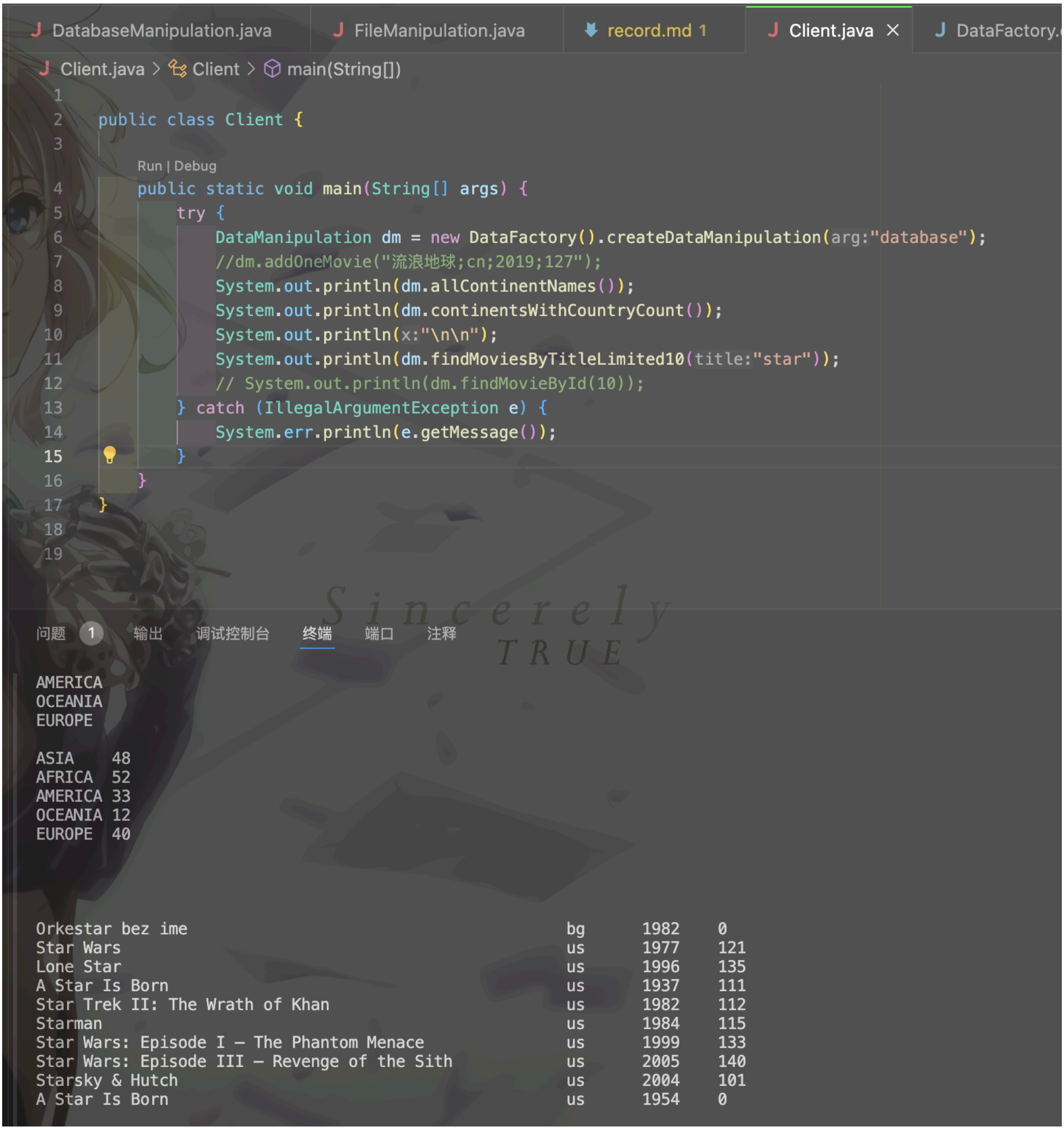


```
[lab3-# \d
                    关联列表
 架构模式 |         名称         |  类型  |   拥有者
----------+----------------------+--------+-----------
 public   | countries            | 数据表 | earendelh
 public   | movies               | 数据表 | earendelh
 public   | movies_movieid_seq   | 序列数 | earendelh
(3 行记录)
```

1. findmoviesbytitlelimite10

database:

```
@Override
  public String findMoviesByTitleLimited10(String title) {
    getConnection();
    StringBuilder sb = new StringBuilder();
    String sql = "select * from movies where lower(title) like '%'||?||'%' limit 10;";
    try {
      PreparedStatement preparedStatement = con.prepareStatement(sql);
      preparedStatement.setString(1, title);
      resultSet = preparedStatement.executeQuery();
      while (resultSet.next()) {
        sb.append(String.format("%-50s", resultSet.getString("title"))+ "\t");
        sb.append(resultSet.getString("country")+ "\t");
        sb.append(resultSet.getInt("year_released") + "\t");
        sb.append(resultSet.getInt("runtime") + "\t");
        sb.append(System.lineSeparator());
      }
    } catch (SQLException e) {
      e.printStackTrace();
    } finally {
      closeConnection();
    }
    return sb.toString();
  }
```

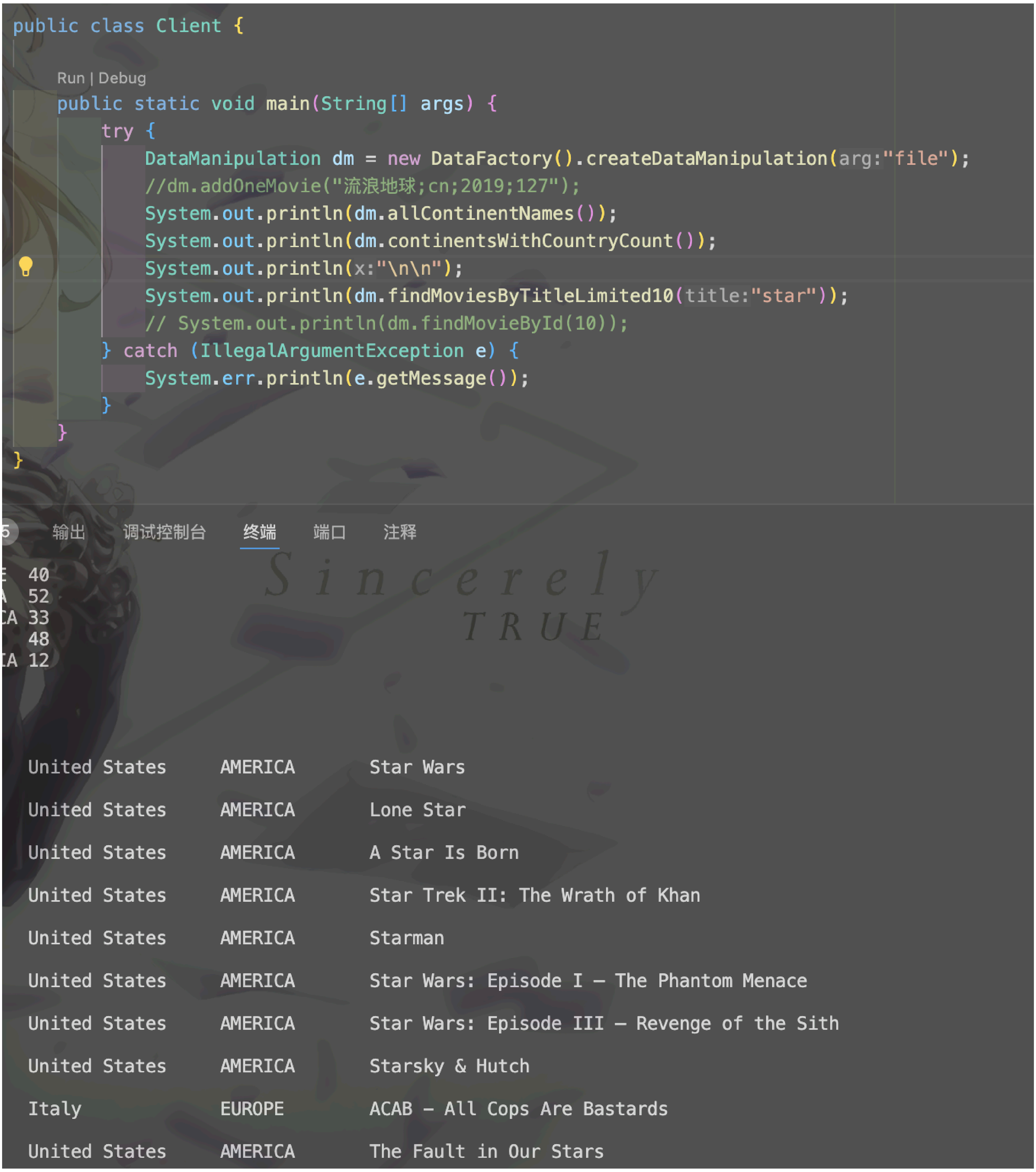


```java

public class Client {

    public static void main(String[] args) {
        try {
            DataManipulation dm = new DataFactory().createDataManipulation(arg:"database");
            //dm.addOneMovie("流浪地球;cn;2019;127");
            System.out.println(dm.allContinentNames());
            System.out.println(dm.continentsWithCountryCount());
            System.out.println(x:"\n\n");
            System.out.println(dm.findMoviesByTitleLimited10(title:"star"));
            // System.out.println(dm.findMovieById(10));
        } catch (IllegalArgumentException e) {
            System.err.println(e.getMessage());
        }
    }
}


```

```
AMERICA
OCEANIA
EUROPE

ASIA    48
AFRICA  52
AMERICA 33
OCEANIA 12
EUROPE  40




Orkestar bez ime                                        bg      1982    0
Star Wars                                               us      1977    121
Lone Star                                               us      1996    135
A Star Is Born                                          us      1937    111
Star Trek II: The Wrath of Khan                         us      1982    112
Starman                                                 us      1984    115
Star Wars: Episode I – The Phantom Menace               us      1999    133
Star Wars: Episode III – Revenge of the Sith            us      2005    140
Starsky & Hutch                                         us      2004    101
A Star Is Born                                          us      1954    0
```

file:

```
@Override
  public String findMoviesByTitleLimited10(String title) {
    Map<String, String> countryMap = getCountryMap();
    List<FullInformation> list = new ArrayList<>();
    String line;
    String[] splitArray;
    int titleIndex = 1, countryIndex = 2, runTimeIndex = 4, runTime;
    int count = 0;
    try (BufferedReader bufferedReader = new BufferedReader(new FileReader("movies.txt"))) {
      bufferedReader.readLine();
      while ((line = bufferedReader.readLine()) != null) {
        splitArray = line.split(";");
        if (splitArray[titleIndex].toLowerCase().contains(title.toLowerCase())) {
          if (!"null".equals(splitArray[runTimeIndex])) {
            runTime = Integer.parseInt(splitArray[runTimeIndex]);
            line = runTime + "\t" + countryMap.get(splitArray[countryIndex].trim()) + "\t"
                + splitArray[titleIndex] + "\n";
            list.add(new FullInformation(runTime, line));
            count++;
          }
        }
        if (count == 10) {
          break;
        }
      }
    } catch (IOException e) {
      e.printStackTrace();
    }
    String result="";
    for(FullInformation f : list) {
      result+=f.information+"\n";

    }
    return result;
  }
```

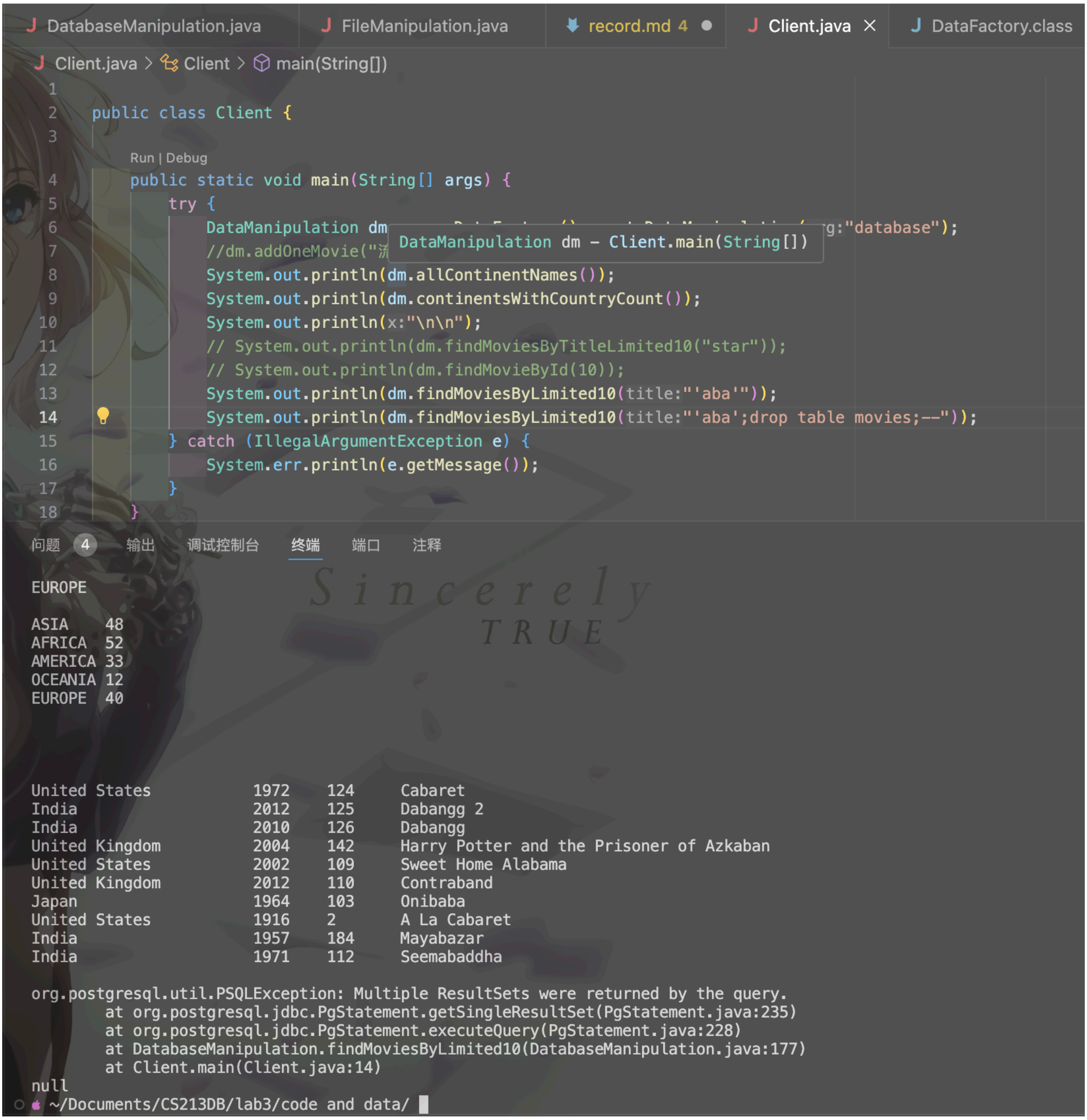


```
public class Client {

    Run | Debug
    public static void main(String[] args) {
        try {
            DataManipulation dm = new DataFactory().createDataManipulation(arg:"file");
            //dm.addOneMovie("流浪地球;cn;2019;127");
            System.out.println(dm.allContinentNames());
            System.out.println(dm.continentsWithCountryCount());
            System.out.println(x:"\n\n");
            System.out.println(dm.findMoviesByTitleLimited10(title:"star"));
            // System.out.println(dm.findMovieById(10));
        } catch (IllegalArgumentException e) {
            System.err.println(e.getMessage());
        }
    }
}
```

```
E  40
A  52
CA 33
   48
IA 12



   United States     AMERICA      Star Wars

   United States     AMERICA      Lone Star

   United States     AMERICA      A Star Is Born

   United States     AMERICA      Star Trek II: The Wrath of Khan

   United States     AMERICA      Starman

   United States     AMERICA      Star Wars: Episode I – The Phantom Menace

   United States     AMERICA      Star Wars: Episode III – Revenge of the Sith

   United States     AMERICA      Starsky & Hutch

   Italy             EUROPE       ACAB – All Cops Are Bastards

   United States     AMERICA      The Fault in Our Stars
```

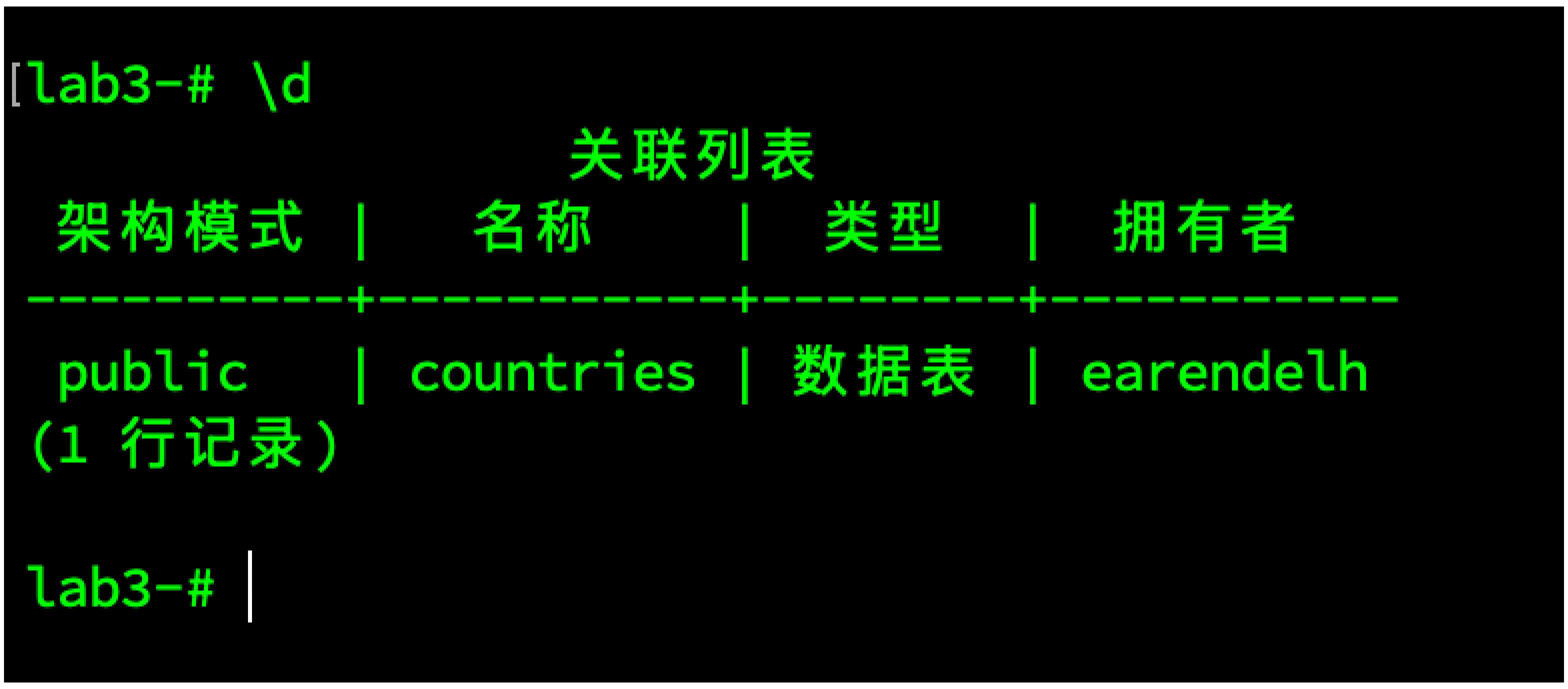
2. sql injection

use statement



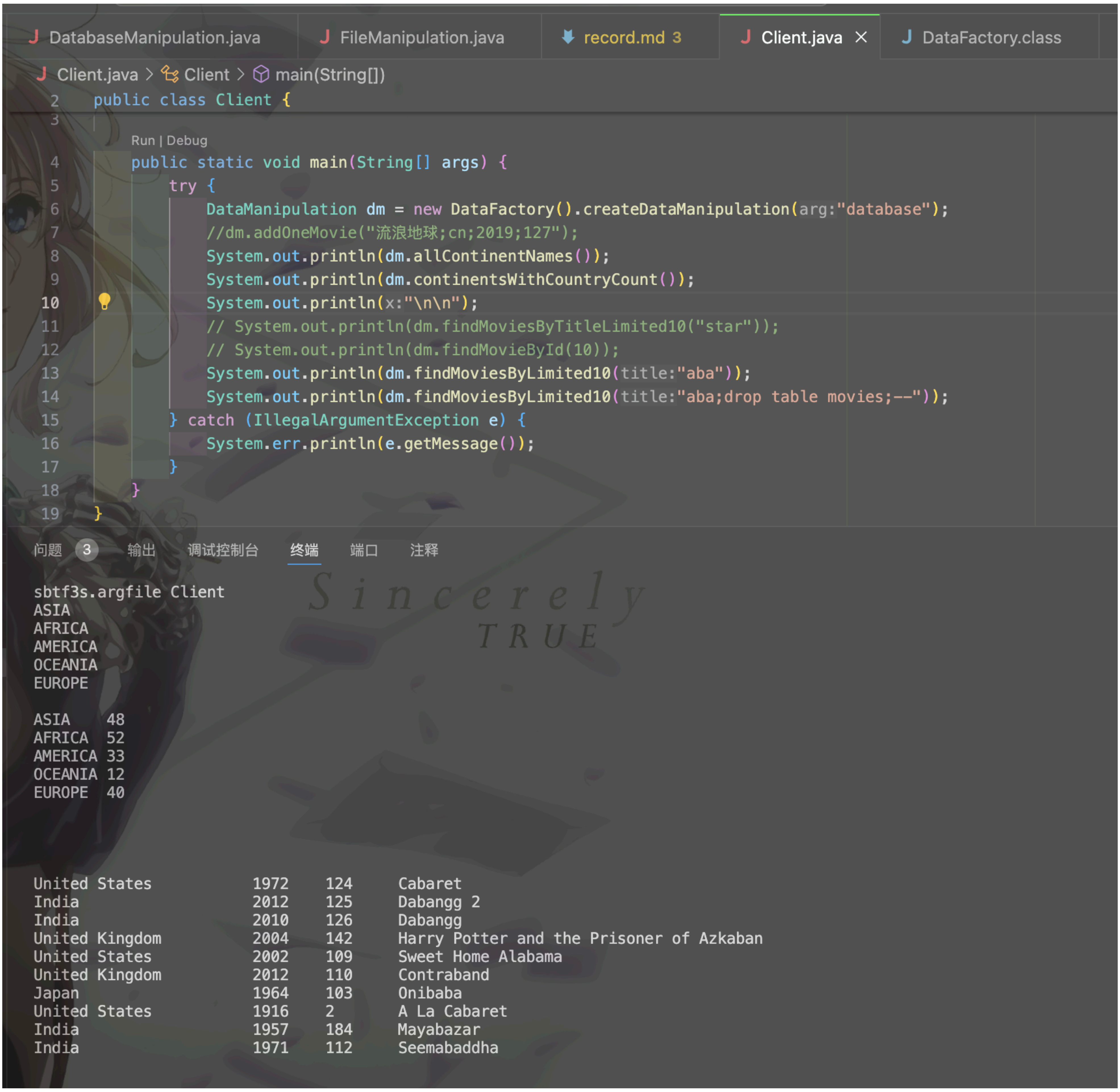
```java

public class Client {

    public static void main(String[] args) {
        try {
            DataManipulation dm                                          g:"database");
            //dm.addOneMovie("浏 DataManipulation dm - Client.main(String[])
            System.out.println(dm.allContinentNames());
            System.out.println(dm.continentsWithCountryCount());
            System.out.println(x:"\n\n");
            // System.out.println(dm.findMoviesByTitleLimited10("star"));
            // System.out.println(dm.findMovieById(10));
            System.out.println(dm.findMoviesByLimited10(title:"'aba'"));
            System.out.println(dm.findMoviesByLimited10(title:"'aba';drop table movies;--"));
        } catch (IllegalArgumentException e) {
            System.err.println(e.getMessage());
        }
    }
```

```
EUROPE

ASIA    48
AFRICA  52
AMERICA 33
OCEANIA 12
EUROPE  40



United States           1972    124     Cabaret
India                   2012    125     Dabangg 2
India                   2010    126     Dabangg
United Kingdom          2004    142     Harry Potter and the Prisoner of Azkaban
United States           2002    109     Sweet Home Alabama
United Kingdom          2012    110     Contraband
Japan                   1964    103     Onibaba
United States           1916    2       A La Cabaret
India                   1957    184     Mayabazar
India                   1971    112     Seemabaddha

org.postgresql.util.PSQLException: Multiple ResultSets were returned by the query.
        at org.postgresql.jdbc.PgStatement.getSingleResultSet(PgStatement.java:235)
        at org.postgresql.jdbc.PgStatement.executeQuery(PgStatement.java:228)
        at DatabaseManipulation.findMoviesByLimited10(DatabaseManipulation.java:177)
        at Client.main(Client.java:14)
null
○ ~/Documents/CS213DB/lab3/code and data/ 
```



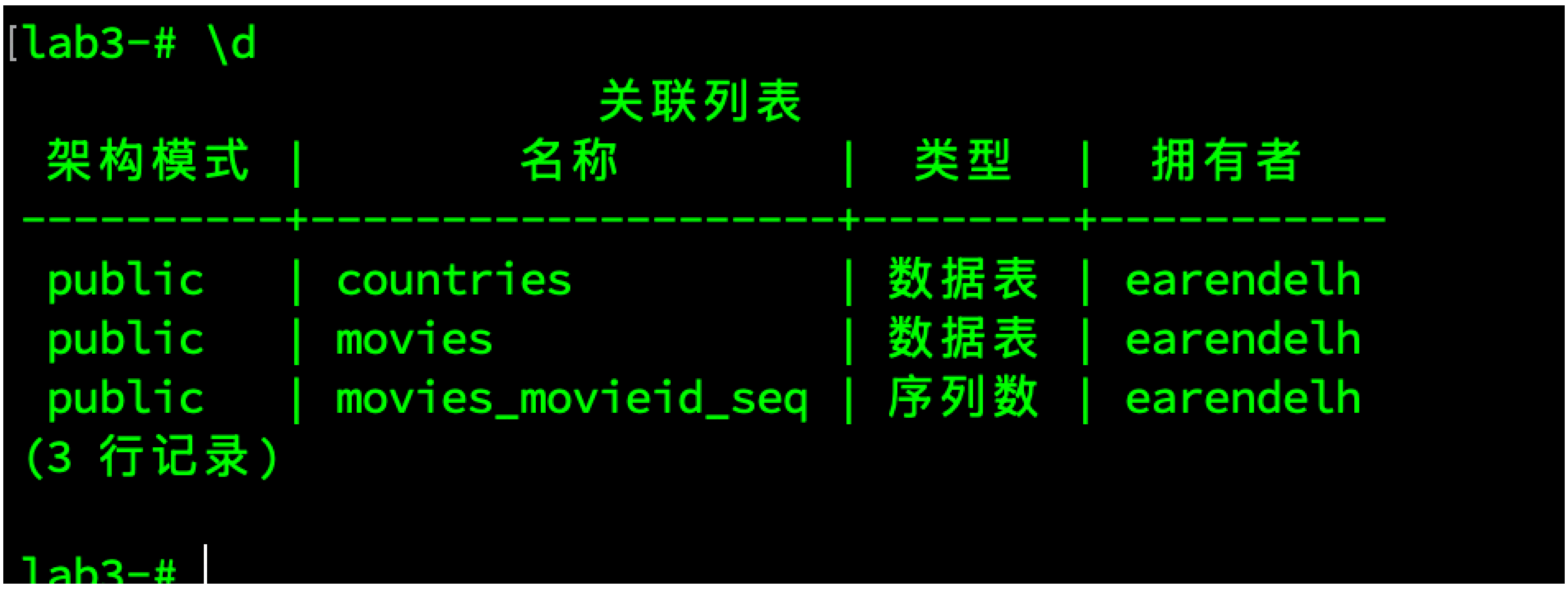
```
[lab3-# \d
                关联列表
 架构模式 |   名称    |  类型  |  拥有者
----------+-----------+--------+-----------
 public   | countries | 数据表 | earendelh
(1 行记录)

lab3-# 
```

use prepared statement



```
public class Client {

    public static void main(String[] args) {
        try {
            DataManipulation dm = new DataFactory().createDataManipulation(arg:"database");
            //dm.addOneMovie("流浪地球;cn;2019;127");
            System.out.println(dm.allContinentNames());
            System.out.println(dm.continentsWithCountryCount());
            System.out.println(x:"\n\n");
            // System.out.println(dm.findMoviesByTitleLimited10("star"));
            // System.out.println(dm.findMovieById(10));
            System.out.println(dm.findMoviesByLimited10(title:"aba"));
            System.out.println(dm.findMoviesByLimited10(title:"aba;drop table movies;--"));
        } catch (IllegalArgumentException e) {
            System.err.println(e.getMessage());
        }
    }
}
```

```
sbtf3s.argfile Client
ASIA
AFRICA
AMERICA
OCEANIA
EUROPE

ASIA    48
AFRICA  52
AMERICA 33
OCEANIA 12
EUROPE  40



United States           1972    124     Cabaret
India                   2012    125     Dabangg 2
India                   2010    126     Dabangg
United Kingdom          2004    142     Harry Potter and the Prisoner of Azkaban
United States           2002    109     Sweet Home Alabama
United Kingdom          2012    110     Contraband
Japan                   1964    103     Onibaba
United States           1916    2       A La Cabaret
India                   1957    184     Mayabazar
India                   1971    112     Seemabaddha
```



```
[lab3-# \d
                    关联列表
 架构模式 |          名称          |  类型  |  拥有者
----------+------------------------+--------+-----------
 public   | countries              | 数据表 | earendelh
 public   | movies                 | 数据表 | earendelh
 public   | movies_movieid_seq     | 序列数 | earendelh
(3 行记录)

lab3-#
```