Import:

```
val mov = spark.read.format("csv").option("delimiter", "\t") .load("/user/verulam_blue/data/movies.tsv")

movies.show(5)
```

```
+-----------------+-------------+----+
|              _c0|          _c1| _c2|
+-----------------+-------------+----+
|McClure, Marc (I)|Freaky Friday|2003|
|McClure, Marc (I)| Coach Carter|2005|
|McClure, Marc (I)|  Superman II|1980|
|McClure, Marc (I)|    Apollo 13|1995|
|McClure, Marc (I)|     Superman|1978|
+-----------------+-------------+----+
only showing top 5 rows
```

```
val movr = spark.read.format("csv").option("delimiter", "\t")
.load("/user/verulam_blue/data/movie-ratings.tsv")
movieratings.show
```

```
+-------+--------------------+----+
|    _c0|                 _c1| _c2|
+-------+--------------------+----+
| 1.6339|'Crocodile' Dunde...|1988|
| 7.6177|                  10|1979|
| 1.2864|10 Things I Hate ...|1999|
| 0.3243|           10,000 BC|2008|
| 0.3376|      101 Dalmatians|1996|
| 0.5218|      102 Dalmatians|2000|
|12.8205|                1066|2012|
| 0.6829|                  12|2007|
| 7.4061|           12 Rounds|2009|
| 2.3677|           127 Hours|2010|
| 1.3585|      13 Going on 30|2004|
| 8.4034|     13 game sayawng|2006|
|   0.59|                1408|2007|
| 4.4292|          15 Minutes|2001|
```

Rename:

```
val movie = mov.withColumnRenamed("_c0", "actor").withColumnRenamed("_c1", "title")
.withColumnRenamed("_c2", "year")
movie.show(5)
```

```
+-----------------+-------------+----+
|            actor|        title|year|
+-----------------+-------------+----+
|McClure, Marc (I)|Freaky Friday|2003|
|McClure, Marc (I)| Coach Carter|2005|
|McClure, Marc (I)|  Superman II|1980|
|McClure, Marc (I)|    Apollo 13|1995|
|McClure, Marc (I)|     Superman|1978|
+-----------------+-------------+----+
only showing top 5 rows
```

```
var movierating = movr.withColumnRenamed("_c0", "rating").withColumnRenamed("_c1", "title")
.withColumnRenamed("_c2", "year")
movierating.show(5)
```

```
+------+--------------------+----+
|rating|               title|year|
+------+--------------------+----+
|1.6339|'Crocodile' Dunde...|1988|
|7.6177|                  10|1979|
|1.2864|10 Things I Hate ...|1999|
|0.3243|           10,000 BC|2008|
|0.3376|      101 Dalmatians|1996|
+------+--------------------+----+
```

Merge:

```
var movie_join = movie.join(movierating, Seq("title", "year"))
movie_join.show(5)
```

```
+-------------+----+-----------------+------+
|        title|year|            actor|rating|
+-------------+----+-----------------+------+
|Freaky Friday|2003|McClure, Marc (I)|0.3847|
| Coach Carter|2005|McClure, Marc (I)|0.9858|
|  Superman II|1980|McClure, Marc (I)|0.8739|
|    Apollo 13|1995|McClure, Marc (I)|1.0267|
|     Superman|1978|McClure, Marc (I)|1.1982|
+-------------+----+-----------------+------+
only showing top 5 rows
```

1.	**Compute the number of movies each actor was in. The output should have two columns: actor, count. The output should be ordered by the count in descending order**

	`val count_actor = movie.groupBy("actor").count().orderBy(desc("count"))`
	`count_actor.show`
	
	+--------------------+-----+
	| actor|count|
	+--------------------+-----+
	| Tatasciore, Fred| 38|
	| Welker, Frank| 38|
	| Jackson, Samuel L.| 32|
	| Harnell, Jess| 31|
	| Willis, Bruce| 27|
	| Damon, Matt| 27|
	| Cummings, Jim (I)| 26|
	| Hanks, Tom| 25|
	| McGowan, Mickie| 25|
	| Lynn, Sherry (I)| 25|
	| Bergen, Bob (I)| 25|
	| Proctor, Phil| 24|
	| Wilson, Owen (I)| 23|
	| Pitt, Brad| 23|
	| Cruise, Tom| 23|
	|Williams, Robin (I)| 22|
	| Morrison, Rana| 22|
	|Freeman, Morgan (I)| 22|
	| Depp, Johnny| 22|
	| De Niro, Robert| 21|
	+--------------------+-----+
	only showing top 20 rows

2.	**Compute the highest-rated movie per year and include all the actors played in that movie. The output should have only one movie per year, and it should contain four columns: year, movie title, rating, a semicolon-separated list of actor names. This question requires a join between *movies.tsv* and *movie-ratings.tsv* files. There are two approaches to this problem. The first is to figure out the highest-rated movies per year and then join with a list of actors. The second one is to perform the join first and then figure out the highest-rated movies per year and a list of actors. The result of each approach is different from the other one. Why do you think that is?**
	First Approach

```
var approach1 = movierating.groupBy("year").agg(max("rating").as("ratings"))
approach1 = approach1.join(movierating, Seq("year")).join(movie, Seq("year", "title"))
approach1 = approach1.groupBy("year", "title", "rating").agg(collect_list("actor").as("actors"))
approach1.orderBy('year).show
```

```
+----+--------------------+------+--------------------+
|year|               title|rating|              actors|
+----+--------------------+------+--------------------+
|1961|One Hundred and O...|0.6726|[Wright, Ben (I),...|
|1967|     The Jungle Book|1.3485|[Wright, Ben (I),...|
|1972|       The Godfather|0.5099|[Russo, Gianni (I...|
|1973|        The Exorcist|0.6581|[Symonds, Robert,...|
|1975|               Jaws| 0.701|[Cheshire, Denise...|
|1977|           Star Wars|0.0807|[Fisher, Carrie, ...|
|1977|Close Encounters ...|1.1686|[Garr, Teri, Dill...|
|1977|Saturday Night Fever|1.2184|[Michaels, Bert (...|
|1978|              Grease| 0.413|[Biehn, Michael, ...|
|1978|              Jaws 2|1.9793|     [Scheider, Roy]|
|1978|            Superman|1.1982|[Ratzenberger, Jo...|
|1979|   Kramer vs. Kramer|0.6595|[Seneca, Joe, Ale...|
|1979|           Moonraker| 1.808|[Morse, Ralph (I)...|
|1979|      Operación Ogro|1.0432|    [Atkine, Féodor]|
|1979|      Apocalypse Now|1.9906|[Ermey, R. Lee, S...|
|1979|               Alien|0.7161|[Stanton, Harry D...|
|1980|         Superman II|0.8739|[Ratzenberger, Jo...|
|1980|The Gods Must Be ...|0.8556|    [Weyers, Marius]|
|1980|Star Wars: Episod...|0.2564|[Fisher, Carrie, ...|
|1981|   Absence of Malice|2.1052|[DiSanti, John, N...|
+----+--------------------+------+--------------------+
only showing top 20 rows
```

Second approach

SPARK JOB FINISHED

```
import org.apache.spark.sql.expressions.Window
val joined_data = movierating.join(movie, Seq("title", "year"), "inner")
val windows = Window.partitionBy("year").orderBy(col("rating").desc)
val ranked_data = joined_data.withColumn("rank", dense_rank().over(windows))
val approach2 = ranked_data.filter(col("rank") === 1).groupBy("year", "title", "rating").agg(concat_ws("; ", collect_list("actor"))
.alias("actors")).orderBy(col("year"))

approach2.show
```

```
+----+--------------------+-------+--------------------+
|year|               title| rating|              actors|
+----+--------------------+-------+--------------------+
|1961|One Hundred and O...| 0.6726|Wickes, Mary; Wri...|
|1967|     The Jungle Book| 1.3485|Howard, Clint; Wr...|
|1972|       The Godfather| 0.5099|Brando, Marlon; K...|
|1973|        The Exorcist| 0.6581|Burstyn, Ellen; M...|
|1975|               Jaws|  0.701|Grossman, Ted (I)...|
|1977|Saturday Night Fever| 1.2184|Dillon, Denny; Tr...|
|1978|              Jaws 2| 1.9793|       Scheider, Roy|
|1979|      Apocalypse Now| 1.9906|Brando, Marlon; H...|
|1980|         Superman II| 0.8739|McClure, Marc (I)...|
|1981|   Absence of Malice| 2.1052|Balaban, Bob; Som...|
|1982|         First Blood| 1.2501|Tamburro, Charles...|
|1983|              Yentl| 1.4011|Streisand, Barbra...|
|1984|      The Terminator|  2.061|Miller, Dick (I);...|
|1985|Kiss of the Spide...|    2.1|       Hurt, William|
|1986|    An American Tail|14.2122|Finnegan, John (I...|
|1987|           Mannequin| 1.8974|Bailey, G.W.; Get...|
|1988|        Child's Play| 1.7632|Gale, Ed; Wilder,...|
|1989|     Lethal Weapon 2| 1.7087|Wilson, Norman D....|
|1990|   Presumed Innocent| 2.0055|Dennehy, Brian; L...|
|1991|                 JFK| 2.0171|Herthum, Harold G...|
+----+--------------------+-------+--------------------+
only showing top 20 rows
```

The reason why the result is different between the two approaches is because in the first approach, you find the highest-rated movies per year first and then join with the list of actors, which might include multiple actors. In the second approach, you join the two datasets first, which results in a single row for each movie-actor combination, and then you find the highest-rated movie per year from the joined data. Compared to the second method where we combined the two tables first, this method automatically filter out all the movies that are only exclusive to only one table, so that when we calculate the maximum rating, it results in a complete table when we print it out.

3.	**Determine which pair of actors worked together most. Working together is defined as appearing in the same movie. The output should have three columns: actor1, actor2, and count. The output should be sorted by the count in descending order. The solution to this question requires doing self-join.**

```
val pairs = movie.as("actor1")
  .join(movie.as("actor2"), $"actor1.title" === $"actor2.title" && $"actor1.year" === $"actor2.year" && $"actor1.actor" < $"actor2.a
  .groupBy($"actor1.actor".as("actor1"), $"actor2.actor".as("actor2"))
  .count()
  .orderBy(desc("count"))

pairs.show()
```

SPARK JOB FINISHED

```
+-----------------+------------------+-----+
|           actor1|            actor2|count|
+-----------------+------------------+-----+
| Lynn, Sherry (I)|   McGowan, Mickie|   23|
|  Bergen, Bob (I)|  Lynn, Sherry (I)|   19|
|  Bergen, Bob (I)|   McGowan, Mickie|   19|
|  Angel, Jack (I)|  Lynn, Sherry (I)|   17|
|  Angel, Jack (I)|   McGowan, Mickie|   17|
|  McGowan, Mickie|       Rabson, Jan|   16|
| Lynn, Sherry (I)|       Rabson, Jan|   16|
|Darling, Jennifer|   McGowan, Mickie|   15|
|    Harnell, Jess|   McGowan, Mickie|   14|
|  Bergen, Bob (I)|     Harnell, Jess|   14|
|  Bergen, Bob (I)|       Rabson, Jan|   14|
| Farmer, Bill (I)|   McGowan, Mickie|   14|
|Sandler, Adam (I)|Schneider, Rob (I)|   14|
|Darling, Jennifer|  Lynn, Sherry (I)|   14|
|  Bergen, Bob (I)|   Bumpass, Rodger|   13|
|    Harnell, Jess|  Lynn, Sherry (I)|   13|
| Farmer, Bill (I)|  Lynn, Sherry (I)|   13|
|  Angel, Jack (I)|   Bergen, Bob (I)|   13|
|  McGowan, Mickie|     Proctor, Phil|   12|
|    Covert, Allen| Sandler, Adam (I)|   12|
+-----------------+------------------+-----+
only showing top 20 rows
```

Nomor 1

```
movies2.createOrReplaceTempView("movies")
movies_rate.createOrReplaceTempView("movie_ratings")
```

```
spark.sql("select actor, count(*) as count from movies group by actor order by count desc").show
```

```
+------------------+-----+
|             actor|count|
+------------------+-----+
|   Tatasciore, Fred|   38|
|      Welker, Frank|   38|
| Jackson, Samuel L.|   32|
|      Harnell, Jess|   31|
|      Willis, Bruce|   27|
|        Damon, Matt|   27|
|  Cummings, Jim (I)|   26|
|         Hanks, Tom|   25|
|    McGowan, Mickie|   25|
|   Lynn, Sherry (I)|   25|
|    Bergen, Bob (I)|   25|
|      Proctor, Phil|   24|
|   Wilson, Owen (I)|   23|
|         Pitt, Brad|   23|
```

Took 11 sec. Last updated by anonymous at October 17 2023, 12:09:05 PM. (outdated)

Nomor 2 Cara 1

```
val res = spark.sql(
    """
    SELECT t.year, t.movie_title, t.max_rating, CONCAT_WS(';', collect_set(mo.actor)) AS actor_names
    FROM (
        SELECT m.year, m.title AS movie_title, MAX(r.rating) AS max_rating
        FROM movie_ratings r
        JOIN movies m
        ON m.title = r.title AND m.year = r.year
        GROUP BY m.year, m.title
    ) t
    JOIN movies mo
    ON mo.year = t.year AND mo.title = t.movie_title
    GROUP BY t.year,t.movie_title,t.max_rating
    ORDER BY t.year

    """
)

res.show
```

```
+----+--------------------+----------+--------------------+
|year|         movie_title|max_rating|         actor_names|
+----+--------------------+----------+--------------------+
|1961|One Hundred and O...|    0.6726|Wickes, Mary;Wrig...|
|1967|     The Jungle Book|    1.3485|Howard, Clint;Wri...|
|1972|       The Godfather|    0.5099|Sivero, Frank (I)...|
|1973|        The Exorcist|    0.6581|Mitchell, Donna (...|
|1975|                Jaws|     0.701|Grossman, Ted (I)...|
|1977|           Star Wars|    0.0807|Fielder, Harry;He...|
```

Nomor 2 Cara 2

```
val joinedData = movies2.join(movies_rate,Seq("title","year"))
val MaxRatedYear = joinedData.groupBy("year").agg(max("rating").alias("rating"))
val MaxRatedMovie = joinedData.join(highestRatedYear,Seq("year","rating"))

MaxRatedMovie.groupBy("year","title","rating").agg(concat_ws(";",collect_list("actor")).alias("actors")).orderBy(asc("year")).show
```

```
+----+--------------------+-------+--------------------+
|year|               title| rating|              actors|
+----+--------------------+-------+--------------------+
|1961|One Hundred and O...| 0.6726|Wickes, Mary;Wrig...|
|1967|     The Jungle Book| 1.3485|Howard, Clint;Wri...|
|1972|       The Godfather| 0.5099|Brando, Marlon;Ke...|
|1973|        The Exorcist| 0.6581|Burstyn, Ellen;Mi...|
|1975|                Jaws|  0.701|Grossman, Ted (I)...|
|1977|Saturday Night Fever| 1.2184|Dillon, Denny;Tra...|
|1978|              Jaws 2| 1.9793|       Scheider, Roy|
|1979|      Apocalypse Now| 1.9906|Brando, Marlon;Ho...|
|1980|         Superman II| 0.8739|McClure, Marc (I)...|
|1981|   Absence of Malice| 2.1052|Balaban, Bob;Somm...|
|1982|         First Blood| 1.2501|Tamburro, Charles...|
|1983|               Yentl| 1.4011|Streisand, Barbra...|
|1984|      The Terminator|  2.061|Miller, Dick (I);...|
|1985|Kiss of the Spide...|    2.1|       Hurt, William|
```

Perbedaan **pendekatan 1** (figure max rating first join last) dan **pendekatan 2** (Join first figure max rating last) :

Perbedaan antara kedua pendekatan ini terletak pada urutan operasinya.

- Pendekatan 1 mengidentifikasi film dengan rating tertinggi per tahun sebelum bergabung dengan informasi aktornya, sedangkan Pendekatan 2 melakukan penggabungan terlebih dahulu dan kemudian menentukan film dengan rating tertinggi per tahun.

- Pendekatan 1 hanya mempertimbangkan film dengan rating tertinggi setiap tahunnya, terlepas dari apakah film tersebut memiliki aktor terkait dalam kumpulan datanya, sedangkan Pendekatan 2 yang melakukan join terlebih dahulu hanya mempertimbangkan film dengan aktor saat menentukan film dengan rating tertinggi setiap tahunnya

- Kesimpulannya, perbedaan hasil antara kedua pendekatan ini karena pendekatan 1 hanya berfokus pada rating film dan kemudian mencoba mencari aktor terkait, sedangkan pendekatan 2 menggabungkan informasi film dan rating dengan informasi aktor sebelum mengidentifikasi film dengan rating tertinggi per tahun

Nomor 3

```
val res2 = spark.sql(
    """
    SELECT a1.actor AS actor1, a2.actor AS actor2, COUNT(*) AS count
FROM movies a1
JOIN movies a2 ON a1.title = a2.title AND a1.actor < a2.actor
GROUP BY a1.actor, a2.actor
ORDER BY count desc
    """
).show
```

```
+----------------+-----------------+-----+
|          actor1|           actor2|count|
+----------------+-----------------+-----+
| Lynn, Sherry (I)|   McGowan, Mickie|   23|
|  Bergen, Bob (I)|  Lynn, Sherry (I)|   19|
|  Bergen, Bob (I)|   McGowan, Mickie|   19|
|  Angel, Jack (I)|  Lynn, Sherry (I)|   17|
|  Angel, Jack (I)|   McGowan, Mickie|   17|
|  McGowan, Mickie|       Rabson, Jan|   16|
| Lynn, Sherry (I)|       Rabson, Jan|   16|
|Darling, Jennifer|   McGowan, Mickie|   15|
|    Harnell, Jess|   McGowan, Mickie|   14|
|  Bergen, Bob (I)|     Harnell, Jess|   14|
|  Bergen, Bob (I)|       Rabson, Jan|   14|
| Farmer, Bill (I)|   McGowan, Mickie|   14|
|Sandler, Adam (I)|Schneider, Rob (I)|   14|
|Darling, Jennifer|  Lynn, Sherry (I)|   14|
```