spark关于join后有重复列的问题（org.apache.spark.sql.AnalysisException: Reference '*' is ambiguous）

楚时邀月　2022-11-03　原文

问题

datafrme提供了强大的JOIN操作,但是在操作的时候，经常发现会碰到重复列的问题。在你不注意的时候，去用相关列做其他操作的时候，就会出现问题！

假如这两个字段同时存在，那么就会报错，如下：
org.apache.spark.sql.AnalysisException: Reference 'key2' is ambiguous

实例

1.创建两个df演示实例

```
1.  val df = sc.parallelize(Array(
2.      ("yuwen", "zhangsan", 80), ("yuwen", "lisi", 90),
    ("shuxue", "zhangsan", 90), ("shuxue", "lisi", 95)
3.  )).toDF("course", "name", "score")
```

显示：df.show()

```
+------+--------+-----+
|course|    name|score|
+------+--------+-----+
| yuwen|zhangsan|   80|
| yuwen|    lisi|   90|
|shuxue|zhangsan|   90|
|shuxue|    lisi|   95|
+------+--------+-----+
```

```
1.  val df2 = sc.parallelize(Array(
2.      ("yuwen", "zhangsan", 90), ("shuxue", "zhangsan", 100)
3.  )).toDF("course", "name", "score")
```

显示：df2.show

```
+------+--------+-----+
|course|    name|score|
+------+--------+-----+
| yuwen|zhangsan|   90|
|shuxue|zhangsan|  100|
+------+--------+-----+
```

关联查询：

```
1.  val joined = df.join(df2, df("cource") === df2("cource")
    && df("name") === df2("name"), "left_outer")
```

结果展示：

```
+------+--------+-----+------+--------+-----+
|course|    name|score|course|    name|score|
+------+--------+-----+------+--------+-----+
| yuwen|    lisi|   90|  null|    null| null|
|shuxue|zhangsan|   90|shuxue|zhangsan|  100|
|shuxue|    lisi|   95|  null|    null| null|
| yuwen|zhangsan|   80| yuwen|zhangsan|   90|
+------+--------+-----+------+--------+-----+
```

这时候问题出现了这个地方出现了三个两两相同的字段，当你在次操作这个字段的时候就出问题了。

```
scala> joined.select("name").show
org.apache.spark.sql.AnalysisException: Reference 'name' is ambiguous, could be: name#10, name#16.;
```

解决问题

1.你可以使用的时候指定你要用哪个df里面的字段

```
1.  joined.select(df("course"),df("name")).show
```

结果：

```
+------+--------+
|course|    name|
+------+--------+
| yuwen|    lisi|
|shuxue|zhangsan|
|shuxue|    lisi|
| yuwen|zhangsan|
+------+--------+
```

2.你可以删除多余的列，在实际情况中你不可能将两张完全一样的表进行关联，一般就几个字段的名字相同，这样你可以删除你不需要的字段

```
1.  joined.drop(df2("name"))
```

结果：

```
+------+--------+-----+------+-----+
|course|    name|score|course|score|
+------+--------+-----+------+-----+
| yuwen|    lisi|   90|  null| null|
|shuxue|zhangsan|   90|shuxue|  100|
|shuxue|    lisi|   95|  null| null|
| yuwen|zhangsan|   80| yuwen|   90|
+------+--------+-----+------+-----+
```