

Script 2

In script2, data is read from two files student_data.csv and major.csv. These tables have a common attribute, i.e. major_id. Student_data table has 25,000 rows and major table has 10 rows. The scripts performs a inner join, full outer join, a left outer join on the two tables on the key 'major_id'.

```
Administrator: C:\Windows\System32\cmd.exe - spark-shell

scala> join_fullouter.explain()
== Physical Plan ==
*(9) Project [coalesce(major_id#121, major_id#10) AS major_id#974, school_id#702, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, major_name#11, course_duration#12]
+- SortMergeJoin [major_id#121], [major_id#10], FullOuter
   :- *(6) Sort [major_id#121 ASC NULLS FIRST], false, 0
   :- +- Exchange hashpartitioning(major_id#121, 200)
   :    +- *(5) Project [coalesce(school_id#120, school_id#39) AS school_id#702, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, major_id#121, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46]
   :       +- SortMergeJoin [school_id#120], [school_id#39], FullOuter
   :          :- *(2) Sort [school_id#120 ASC NULLS FIRST], false, 0
   :          :- +- Exchange hashpartitioning(school_id#120, 200)
   :              +- *(1) FileScan csv [stud_id#111,first_name#112,last_name#113,birthday#114,gender#115,age#116,stud_email#117,stud_phone#118,stud_address#119,school_id#120,major_id#121] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/student_data.csv], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<stud_id:string,first_name:string,last_name:string,birthday:string,gender:string,age:string...
   :              +- *(4) Sort [school_id#39 ASC NULLS FIRST], false, 0
   :              +- Exchange hashpartitioning(school_id#39, 200)
   :                  +- *(3) FileScan csv [school_id#39,school_name#40,school_email#41,school_phone#42,street#43,city#44,state#45,zip_code#46] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/school_data.csv], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<school_id:string,school_name:string,school_email:string,school_phone:string,street:string,city:string,state:string,zip_code:string...
   :              ...
   :          +- *(8) Sort [major_id#10 ASC NULLS FIRST], false, 0
   :          +- Exchange hashpartitioning(major_id#10, 200)
   :              +- *(7) FileScan csv [major_id#10,major_name#11,course_duration#12] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/major.csv], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<major_id:string,major_name:string,course_duration:string>

scala> join_right.explain()
== Physical Plan ==
*(6) Project [major_id#10, school_id#39, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, major_name#11, course_duration#12]
+- SortMergeJoin [major_id#121], [major_id#10], RightOuter
   :- *(3) Sort [major_id#121 ASC NULLS FIRST], false, 0
```

In the above screen shot , the physical plan of full outer join where Sort Merge method is used is shown.

```
Administrator: C:\Windows\System32\cmd.exe - spark-shell
scala> spark.time(df3.join(df2,Seq("school_id"),"left").join(df1,Seq("major_id"),"left"))
Time taken: 20 ms
res25: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> join_inner.explain()
== Physical Plan ==
*(3) Project [major_id#121, school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, major_name#11, course_duration#12]
+- *(3) BroadcastHashJoin [major_id#121], [major_id#10], Inner, BuildRight
   :- *(3) Project [school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, major_id#121, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46]
      : +- *(3) BroadcastHashJoin [school_id#120], [school_id#39], Inner, BuildRight
         : +- *(3) Project [stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_id#120, major_id#121]
            : +- *(3) Filter (isnotnull(school_id#120) && isnotnull(major_id#121))
               : +- *(3) FileScan csv [stud_id#111,first_name#112,last_name#113,birthday#114,gender#115,age#116,stud_email#117,stud_phone#118,stud_address#119,school_id#120,major_id#121] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/student_data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(school_id)], ReadSchema: struct<stud_id:string,first_name:string,last_name:string,birthday:string,gender:string,age:string...
                  : +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
                     : +- *(1) Project [school_id#39, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46]
                        : +- *(1) Filter isnotnull(school_id#39)
                           : +- *(1) FileScan csv [school_id#39,school_name#40,school_email#41,school_phone#42,street#43,city#44,state#45,zip_code#46] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/school_data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(school_id)], ReadSchema: struct<school_id:string,school_name:string,school_email:string,school_phone:string,street:string,...
                              +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
                                 +- *(2) Project [major_id#10, major_name#11, course_duration#12]
                                    +- *(2) Filter isnotnull(major_id#10)
                                       +- *(2) FileScan csv [major_id#10,major_name#11,course_duration#12] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/major.csv], PartitionFilters: [], PushedFilters: [IsNotNull(major_id)], ReadSchema: struct<major_id:string,major_name:string,course_duration:string>

scala> join_ullouter.explain().explain()
<console>:24: error: not found: value join_ullouter
```

The screen shot above shows the query plan for inner join where Broadcast join is used.

```
Administrator: C:\Windows\System32\cmd.exe - spark-shell
scala> join_left.explain()
== Physical Plan ==
*(3) Project [major_id#121, school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, major_name#11, course_duration#12]
+- *(3) BroadcastHashJoin [major_id#121], [major_id#10], LeftOuter, BuildRight
   :- *(3) Project [school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, major_id#121, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46]
      : +- *(3) BroadcastHashJoin [school_id#120], [school_id#39], LeftOuter, BuildRight
         : +- *(3) FileScan csv [stud_id#111,first_name#112,last_name#113,birthday#114,gender#115,age#116,stud_email#117,stud_phone#118,stud_address#119,school_id#120,major_id#121] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/p/student_data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(school_id)], ReadSchema: struct<stud_id:string,first_name:string,last_name:string,birthday:string,gender:string,age:string...
            : +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
               : +- *(1) Project [school_id#39, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46]
                  : +- *(1) Filter isnotnull(school_id#39)
                     : +- *(1) FileScan csv [school_id#39,school_name#40,school_email#41,school_phone#42,street#43,city#44,state#45,zip_code#46] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/school_data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(school_id)], ReadSchema: struct<school_id:string,school_name:string,school_email:string,school_phone:string,street:string,...
                        +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
                           +- *(2) Project [major_id#10, major_name#11, course_duration#12]
                              +- *(2) Filter isnotnull(major_id#10)
                                 +- *(2) FileScan csv [major_id#10,major_name#11,course_duration#12] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/major.csv], PartitionFilters: [], PushedFilters: [IsNotNull(major_id)], ReadSchema: struct<major_id:string,major_name:string,course_duration:string>

scala>
```

The screenshot above shows the plan for left outer join where Broadcast join is used.

```

Administrator: C:\Windows\System32\cmd.exe - spark-shell
+- *(7) FileScan csv [major_id#10,major_name#11,course_duration#12] Batched: false, Format: CSV, Location: InMemoryFileIndex[
file:/Users/Administrator/Desktop/major.csv], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<major_id:string,major_name:s
tring,course_duration:string>

scala> join_right.explain()
== Physical Plan ==
*(6) Project [major_id#10, school_id#39, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117
, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, major
_name#11, course_duration#12]
+- SortMergeJoin [major_id#121], [major_id#10], RightOuter
  :- *(3) Sort [major_id#121 ASC NULLS FIRST], false, 0
  : +- Exchange hashpartitioning(major_id#121, 200)
  :   +- *(2) Project [school_id#39, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117,
stud_phone#118, stud_address#119, major_id#121, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_c
ode#46]
  :     +- *(2) BroadcastHashJoin [school_id#120], [school_id#39], Inner, BuildRight
  :       :- *(2) Project [stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_ph
one#118, stud_address#119, school_id#120, major_id#121]
  :         +- *(2) Filter (isnotnull(school_id#120) && isnotnull(major_id#121))
  :           +- *(2) FileScan csv [stud_id#111,first_name#112,last_name#113,birthday#114,gender#115,age#116,stud_email#117,stu
d_phone#118,stud_address#119,school_id#120,major_id#121] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administ
rator/Desktop/student_data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(school_id), IsNotNull(major_id)], ReadSchema: struct<
stud_id:string,first_name:string,last_name:string,birthday:string,gender:string,age:string...
  :             +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, string, true]))
  :               +- *(1) FileScan csv [school_id#39,school_name#40,school_email#41,school_phone#42,street#43,city#44,state#45,zip_cod
e#46] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/Users/Administrator/Desktop/school_data.csv], PartitionFilters: [
], PushedFilters: [], ReadSchema: struct<school_id:string,school_name:string,school_email:string,school_phone:string,street:string,...

  +- *(5) Sort [major_id#10 ASC NULLS FIRST], false, 0
  +- Exchange hashpartitioning(major_id#10, 200)
  +- *(4) FileScan csv [major_id#10,major_name#11,course_duration#12] Batched: false, Format: CSV, Location: InMemoryFileIndex[
file:/Users/Administrator/Desktop/major.csv], PartitionFilters: [], PushedFilters: [], ReadSchema: struct<major_id:string,major_name:s
tring,course_duration:string>

scala> join_left.explain()
== Physical Plan ==
*(3) Project [major_id#121, school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#1
17, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, maj

```

The screen shot above explains the plan for right outer join with Sort Merge Join.

Performance Tuning:

Administrator: C:\Windows\System32\cmd.exe - spark-shell

```
scala> val join_inner=df3.join(df2,Seq("school_id"),"inner").join(df1,Seq("major_id"),"inner")
join_inner: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_fullouter=df3.join(df2,Seq("school_id"),"fullouter").join(df1,Seq("major_id"),"fullouter")
join_fullouter: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_right=df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right")
<console>:29: error: value Seq is not a member of org.apache.spark.sql.DataFrame
    val join_right=df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right")
                                ^

scala> val join_right=df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right")
join_right: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_left=df3.join(df2,Seq("school_id"),"left").join(df1,Seq("major_id"),"left")
join_left: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"inner").join(df1,Seq("major_id"),"inner"))
Time taken: 23 ms
res22: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"fullouter").join(df1,Seq("major_id"),"fullouter"))
Time taken: 23 ms
res23: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right"))
Time taken: 17 ms
res24: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"left").join(df1,Seq("major_id"),"left"))
Time taken: 20 ms
res25: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> join_inner.explain()
== Physical Plan ==
*(3) Project [major_id#121, school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, maj
```

Select Administrator: C:\Windows\System32\cmd.exe - spark-shell

```
... 49 elided

scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold").toInt
res33: Int = 100000

scala> spark.conf.set("spark.sql.autoBroadcastJoinThreshold",-1)

scala> val join_inner=df3.join(df2,Seq("school_id"),"inner").join(df1,Seq("major_id"),"inner")
join_inner: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_fullouter=df3.join(df2,Seq("school_id"),"fullouter").join(df1,Seq("major_id"),"fullouter")
join_fullouter: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_right=df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right")
join_right: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> val join_left=df3.join(df2,Seq("school_id"),"left").join(df1,Seq("major_id"),"left")
join_left: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"inner").join(df1,Seq("major_id"),"inner"))
Time taken: 26 ms
res35: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"fullouter").join(df1,Seq("major_id"),"fullouter"))
Time taken: 17 ms
res36: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"right").join(df1,Seq("major_id"),"right"))
Time taken: 16 ms
res37: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> spark.time(df3.join(df2,Seq("school_id"),"left").join(df1,Seq("major_id"),"left"))
Time taken: 11 ms
res38: org.apache.spark.sql.DataFrame = [major_id: string, school_id: string ... 18 more fields]

scala> join_inner.explain()
== Physical Plan ==
*(9) Project [major_id#121, school_id#120, stud_id#111, first_name#112, last_name#113, birthday#114, gender#115, age#116, stud_email#117, stud_phone#118, stud_address#119, school_name#40, school_email#41, school_phone#42, street#43, city#44, state#45, zip_code#46, maj
```

Sr. No	Join Type	Broad Cast Join	Sort Merge Join
1	Inner	23 ms	26 ms
2	Full outer	23ms	17ms
3	Right Outer	17ms	16ms
4	Left Outer	20ms	11ms

Here we can say that the output time of the join varies for different joins . In case the table size for the joins are almost equal , a sort merge join is preferred over Broad Cast join. If there is a comparable difference between the sizes of the table then Broad Cast join is preferred .