**Football – Soccer Dataset Analysis**

[edureka\_918210@ip-20-0-41-164 ~]$ hadoop fs -ls /bigdatapgp/common\_folder/ect3/soccer\_dataset

Found 7 items

-rw-r--r-- 3 evaluationuser01 supergroup 171 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Country.csv

-rw-r--r-- 3 evaluationuser01 supergroup 378 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/League.csv

-rw-r--r-- 3 evaluationuser01 supergroup 279237963 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Match.csv

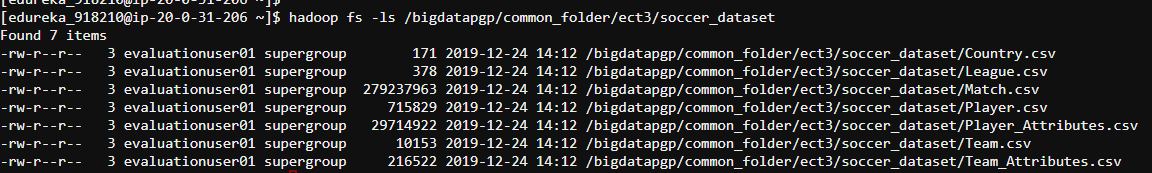
-rw-r--r-- 3 evaluationuser01 supergroup 715829 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Player.csv

-rw-r--r-- 3 evaluationuser01 supergroup 29714922 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Player\_Attributes.csv

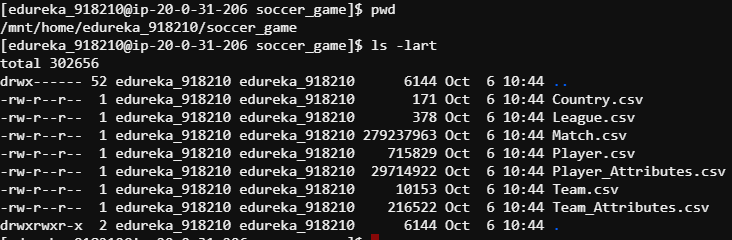
-rw-r--r-- 3 evaluationuser01 supergroup 10153 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Team.csv

-rw-r--r-- 3 evaluationuser01 supergroup 216522 2019-12-24 14:12 /bigdatapgp/common\_folder/ect3/soccer\_dataset/Team\_Attributes.csv

[edureka\_918210@ip-20-0-41-164 ~]$

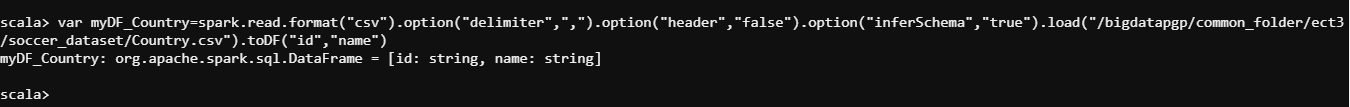


**Downloaded the DataSets from the HDFS to the Local System**

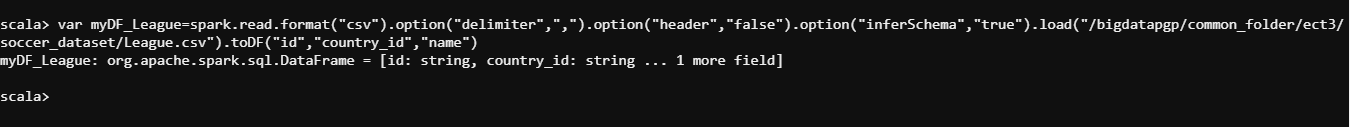


**Spark DF Conversion**

var myDF\_Country=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Country.csv").toDF("id","name")

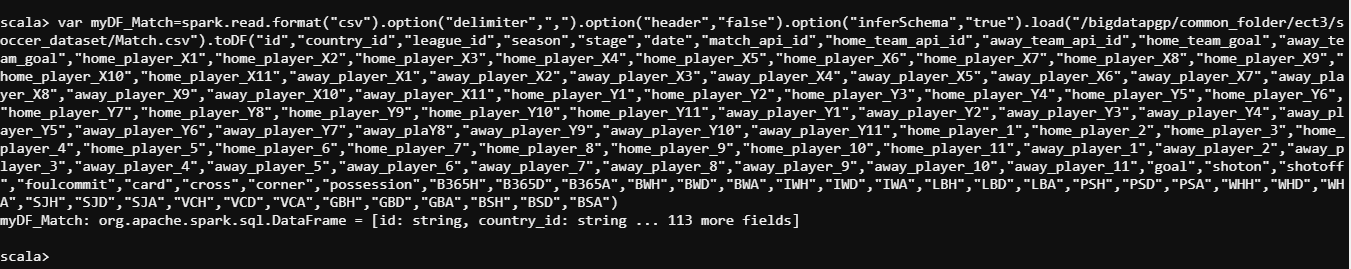


var myDF\_League=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/League.csv").toDF("id","country\_id","name")

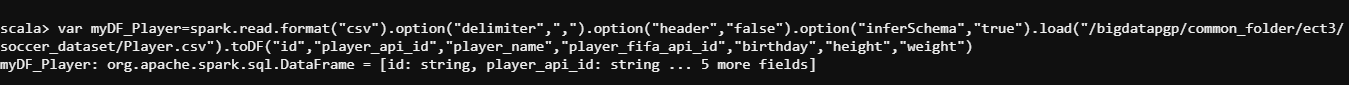


var myDF\_Match=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Match.csv").toDF(

"id","country\_id","league\_id","season","stage","date","match\_api\_id","home\_team\_api\_id","away\_team\_api\_id","home\_team\_goal","away\_team\_goal","home\_player\_X1","home\_player\_X2","home\_player\_X3","home\_player\_X4","home\_player\_X5","home\_player\_X6","home\_player\_X7","home\_player\_X8","home\_player\_X9","home\_player\_X10","home\_player\_X11","away\_player\_X1","away\_player\_X2","away\_player\_X3","away\_player\_X4","away\_player\_X5","away\_player\_X6","away\_player\_X7","away\_player\_X8","away\_player\_X9","away\_player\_X10","away\_player\_X11","home\_player\_Y1","home\_player\_Y2","home\_player\_Y3","home\_player\_Y4","home\_player\_Y5","home\_player\_Y6","home\_player\_Y7","home\_player\_Y8","home\_player\_Y9","home\_player\_Y10","home\_player\_Y11","away\_player\_Y1","away\_player\_Y2","away\_player\_Y3","away\_player\_Y4","away\_player\_Y5","away\_player\_Y6","away\_player\_Y7","away\_player\_Y8","away\_player\_Y9","away\_player\_Y10","away\_player\_Y11","home\_player\_1","home\_player\_2","home\_player\_3","home\_player\_4","home\_player\_5","home\_player\_6","home\_player\_7","home\_player\_8","home\_player\_9","home\_player\_10","home\_player\_11","away\_player\_1","away\_player\_2","away\_player\_3","away\_player\_4","away\_player\_5","away\_player\_6","away\_player\_7","away\_player\_8","away\_player\_9","away\_player\_10","away\_player\_11","goal","shoton","shotoff","foulcommit","card","cross","corner","possession","B365H","B365D","B365A","BWH","BWD","BWA","IWH","IWD","IWA","LBH","LBD","LBA","PSH","PSD","PSA","WHH","WHD","WHA","SJH","SJD","SJA","VCH","VCD","VCA","GBH","GBD","GBA","BSH","BSD","BSA")

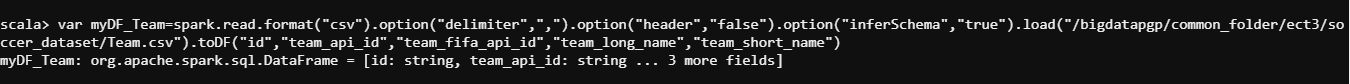


var myDF\_Player=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Player.csv").toDF("id","player\_api\_id","player\_name","player\_fifa\_api\_id","birthday","height","weight")

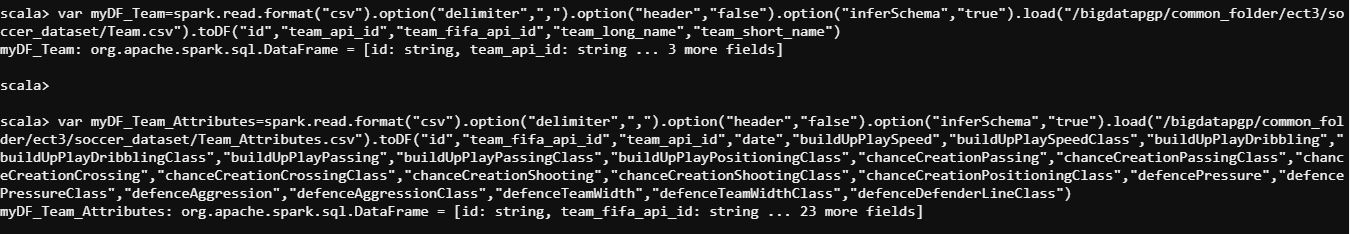


var myDF\_Player\_Attributes=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Player\_Attributes.csv").toDF("id","player\_fifa\_api\_id","player\_api\_id","date","overall\_rating","potential","preferred\_foot","attacking\_work\_rate","defensive\_work\_rate","crossing","finishing","heading\_accuracy","short\_passing,volleys","dribbling","curve","free\_kick\_accuracy","long\_passing","ball\_control","acceleration","sprint\_speed","agility","reactions","balance","shot\_power","jumping","stamina","strength","long\_shots","aggression","interceptions","positioning","vision","penalties","marking","standing\_tackle","sliding\_tackle","gk\_diving","gk\_handling","gk\_kicking","gk\_positioning","gk\_reflexes")

var myDF\_Team=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Team.csv").toDF("id","team\_api\_id","team\_fifa\_api\_id","team\_long\_name","team\_short\_name")



var myDF\_Team\_Attributes=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Team\_Attributes.csv").toDF("id","team\_fifa\_api\_id","team\_api\_id","date","buildUpPlaySpeed","buildUpPlaySpeedClass","buildUpPlayDribbling","buildUpPlayDribblingClass","buildUpPlayPassing","buildUpPlayPassingClass","buildUpPlayPositioningClass","chanceCreationPassing","chanceCreationPassingClass","chanceCreationCrossing","chanceCreationCrossingClass","chanceCreationShooting","chanceCreationShootingClass","chanceCreationPositioningClass","defencePressure","defencePressureClass","defenceAggression","defenceAggressionClass","defenceTeamWidth","defenceTeamWidthClass","defenceDefenderLineClass")



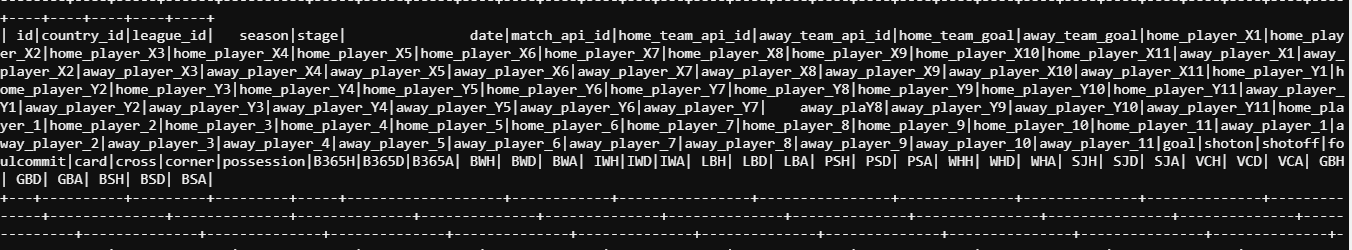
myDF\_Country.show(false)



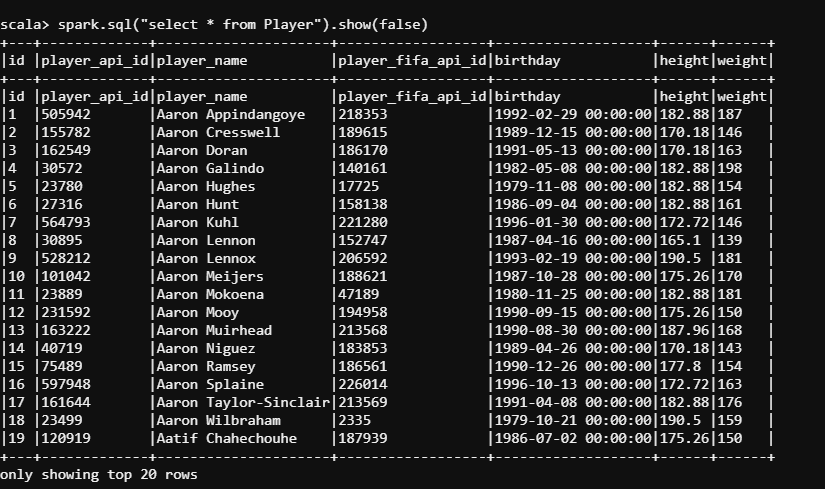
myDF\_League.show(false)



myDF\_Match.show(false)

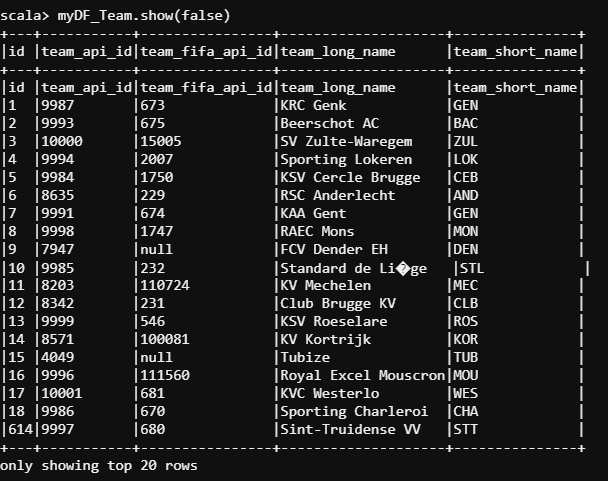


myDF\_Player.show(false)

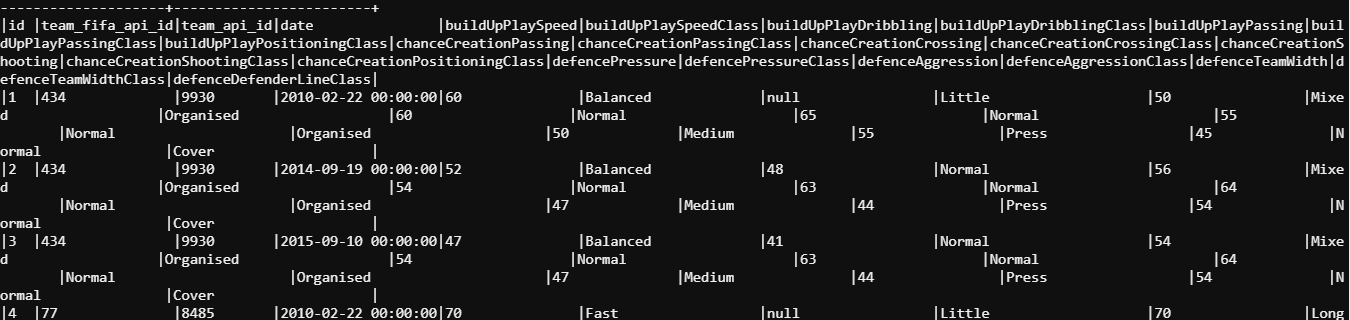


myDF\_Player\_Attributes.show(false)

myDF\_Team.show(false)



myDF\_Team\_Attributes.show(false)



myDF\_Country.createOrReplaceTempView("Country")

myDF\_League.createOrReplaceTempView("League")

myDF\_Match.createOrReplaceTempView("Match")

myDF\_Player.createOrReplaceTempView("Player")

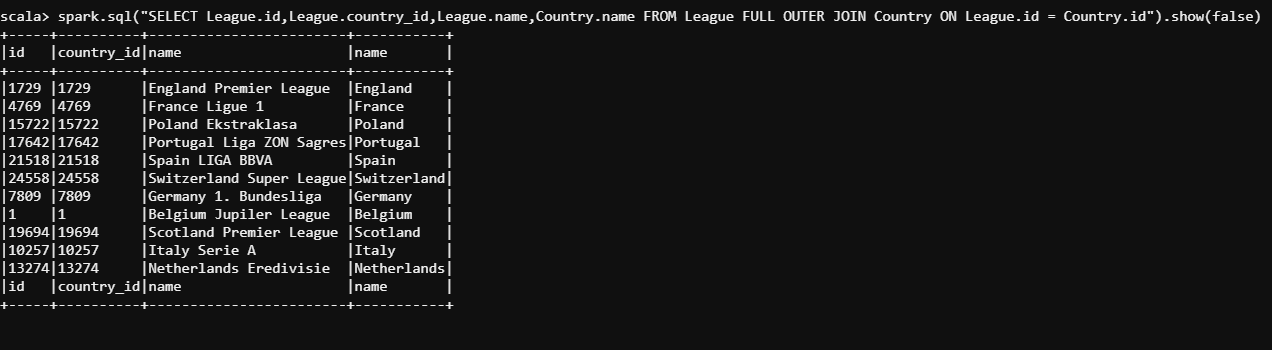
myDF\_Player\_Attributes.createOrReplaceTempView("Player\_Attributes")

myDF\_Team.createOrReplaceTempView("Team")

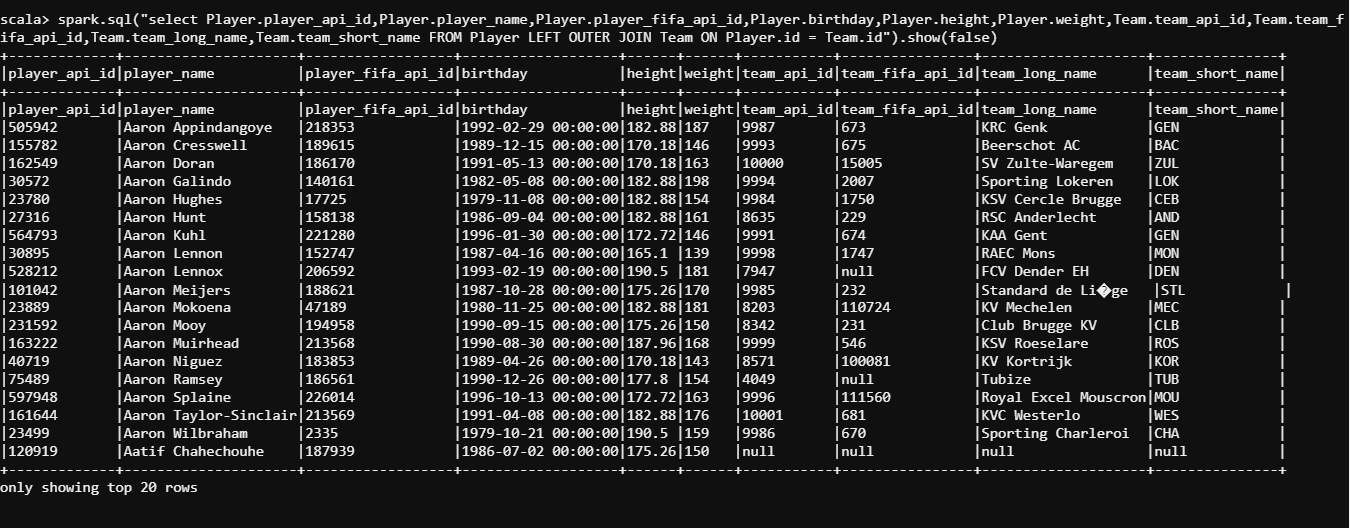
myDF\_Team\_Attributes.createOrReplaceTempView("Team\_Attributes")

**SPARK SQL**

spark.sql("SELECT League.id,League.country\_id,League.name,Country.name FROM League FULL OUTER JOIN Country ON League.id = Country.id").show(false)

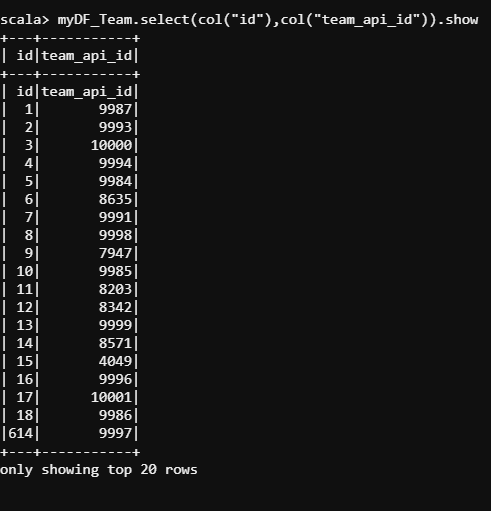


spark.sql("selectPlayer.player\_api\_id,Player.player\_name,Player.player\_fifa\_api\_id,Player.birthday,Player.height,Player.weight,Team.team\_api\_id,Team.team\_fifa\_api\_id,Team.team\_long\_name,Team.team\_short\_name FROM Player LEFT OUTER JOIN Team ON Player.id = Team.id").show(false)

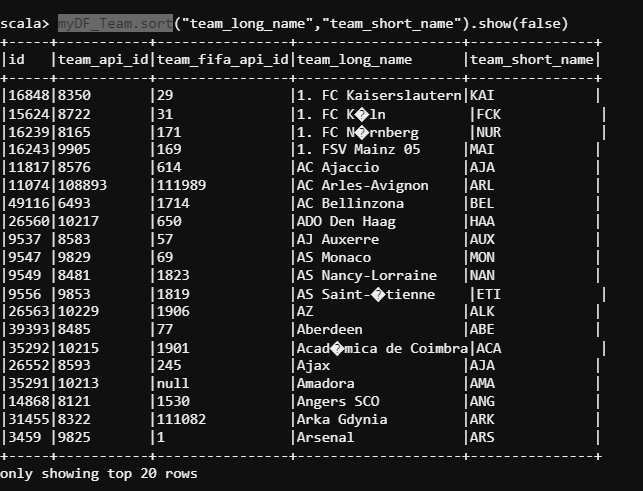


**DF USE**

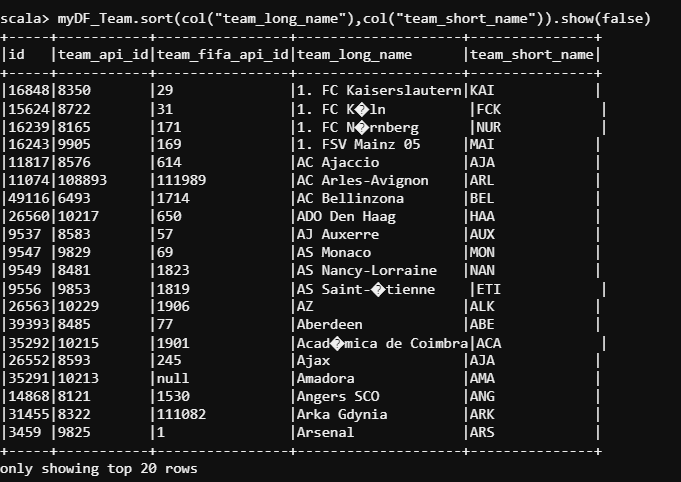
myDF\_Team.select(col("id"),col("team\_api\_id")).show



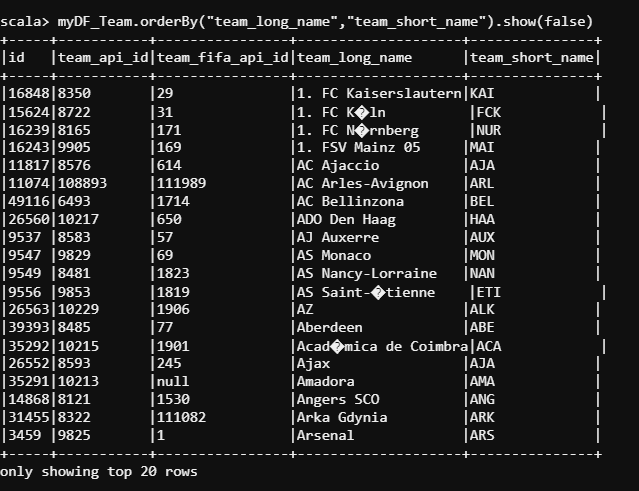
myDF\_Team.sort("team\_long\_name","team\_short\_name").show(false)



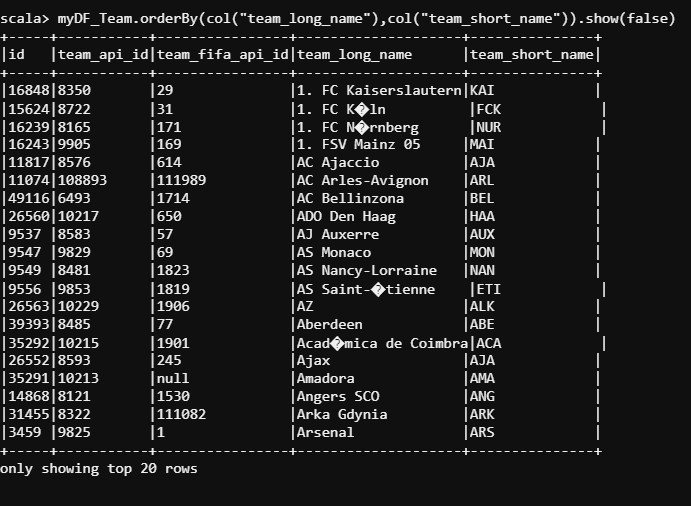
myDF\_Team.sort(col("team\_long\_name"),col("team\_short\_name")).show(false)



myDF\_Team.orderBy("team\_long\_name","team\_short\_name").show(false)



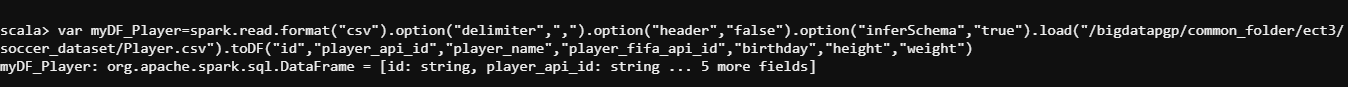
myDF\_Team.orderBy(col("team\_long\_name"),col("team\_short\_name")).show(false)



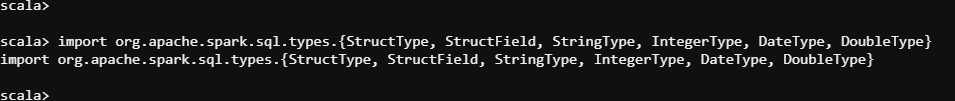
CAST COLUMN NAME TYPE

============================

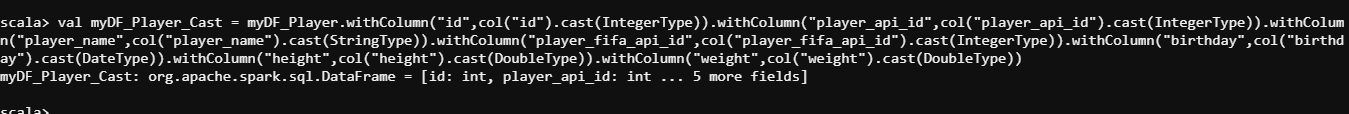
var myDF\_Player=spark.read.format("csv").option("delimiter",",").option("header","false").option("inferSchema","true").load("/bigdatapgp/common\_folder/ect3/soccer\_dataset/Player.csv").toDF("id","player\_api\_id","player\_name","player\_fifa\_api\_id","birthday","height","weight")



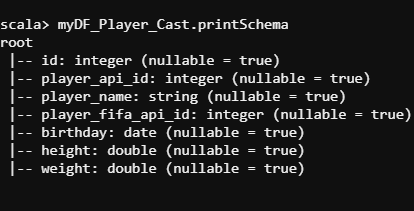
import org.apache.spark.sql.types.{StructType, StructField, StringType, IntegerType, DateType, DoubleType};



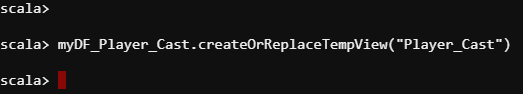
val myDF\_Player\_Cast = myDF\_Player.withColumn("id",col("id").cast(IntegerType)).withColumn("player\_api\_id",col("player\_api\_id").cast(IntegerType)).withColumn("player\_name",col("player\_name").cast(StringType)).withColumn("player\_fifa\_api\_id",col("player\_fifa\_api\_id").cast(IntegerType)).withColumn("birthday",col("birthday").cast(DateType)).withColumn("height",col("height").cast(DoubleType)).withColumn("weight",col("weight").cast(DoubleType))



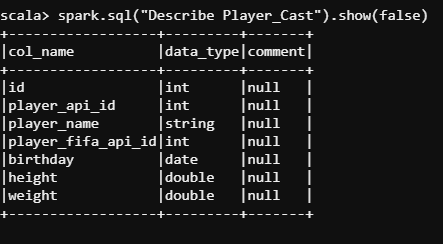
myDF\_Player\_Cast.printSchema



myDF\_Player\_Cast.createOrReplaceTempView("Player\_Cast")



spark.sql("Describe Player\_Cast").show(false)

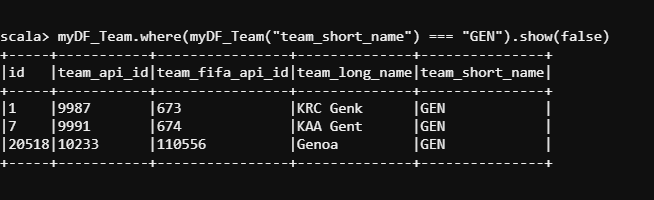


WHERE EXAMPLES IN DF

============================

myDF\_Team.show(false)

myDF\_Team.where(myDF\_Team("team\_short\_name") === "GEN").show(false)

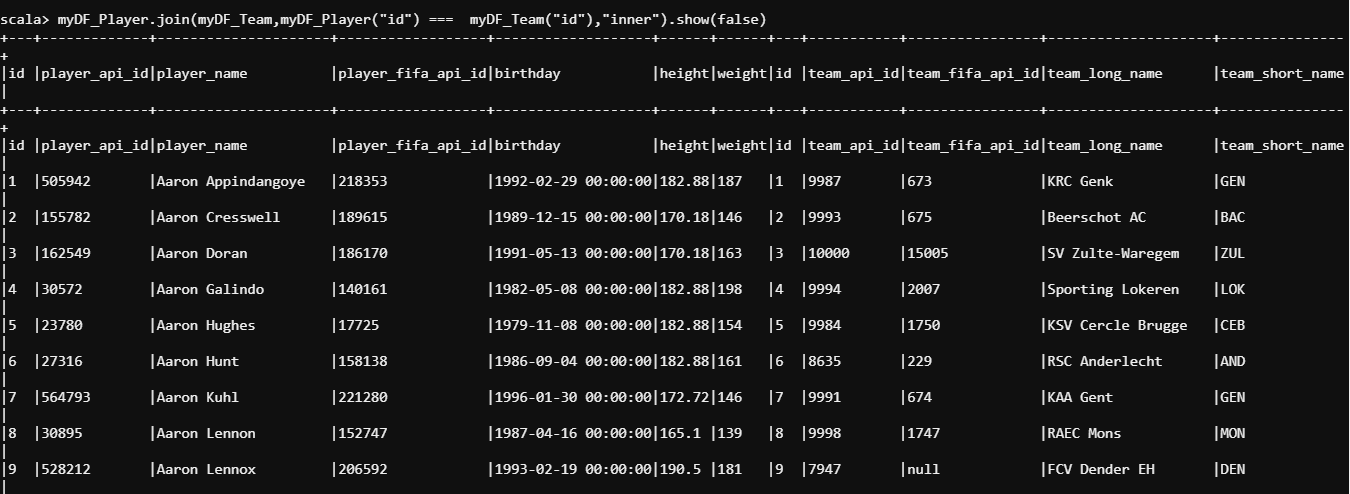


JOINING IN DF

===============

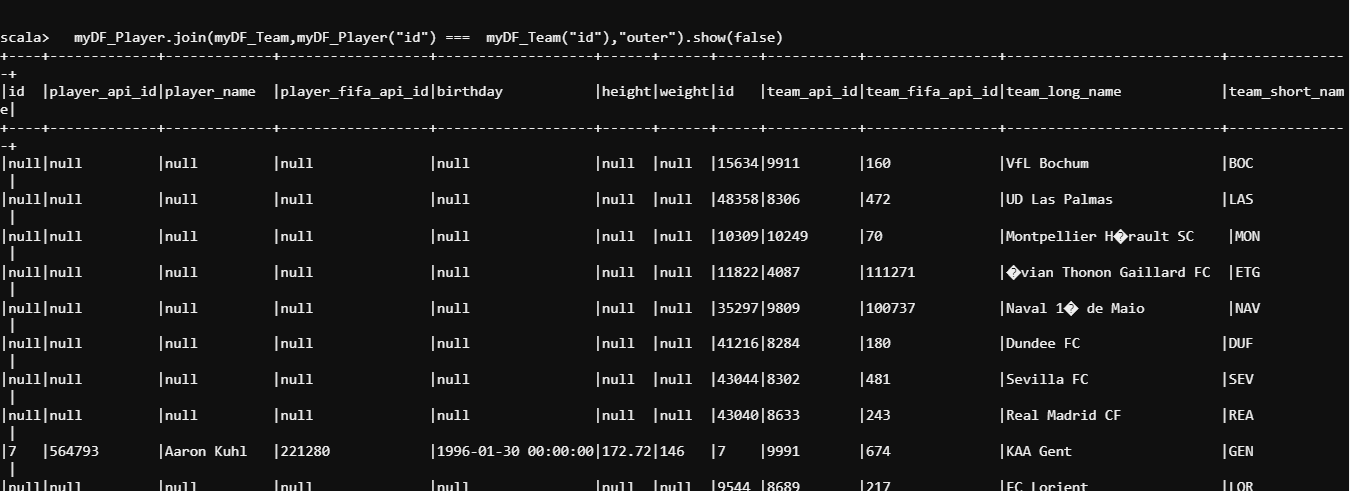
println("Inner join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"inner").show(false)



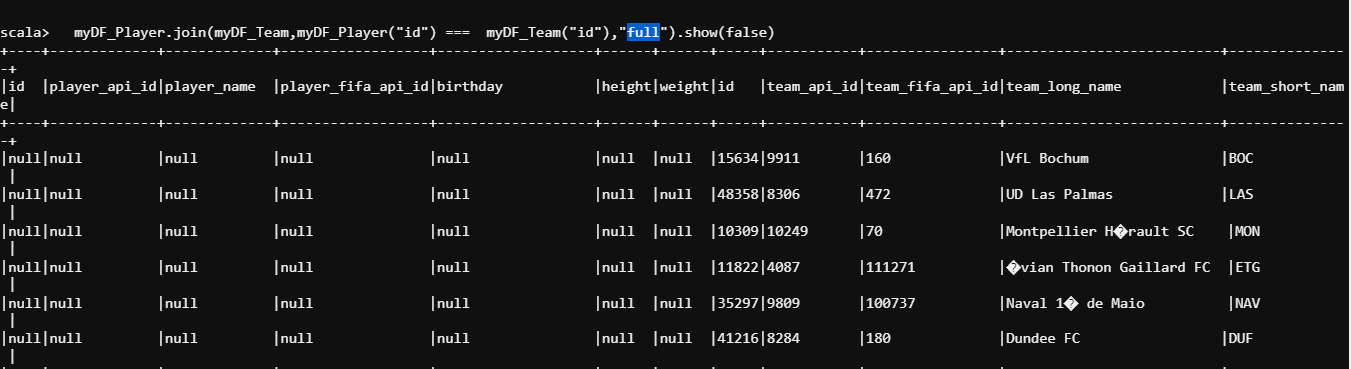
println("Outer join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"outer").show(false)



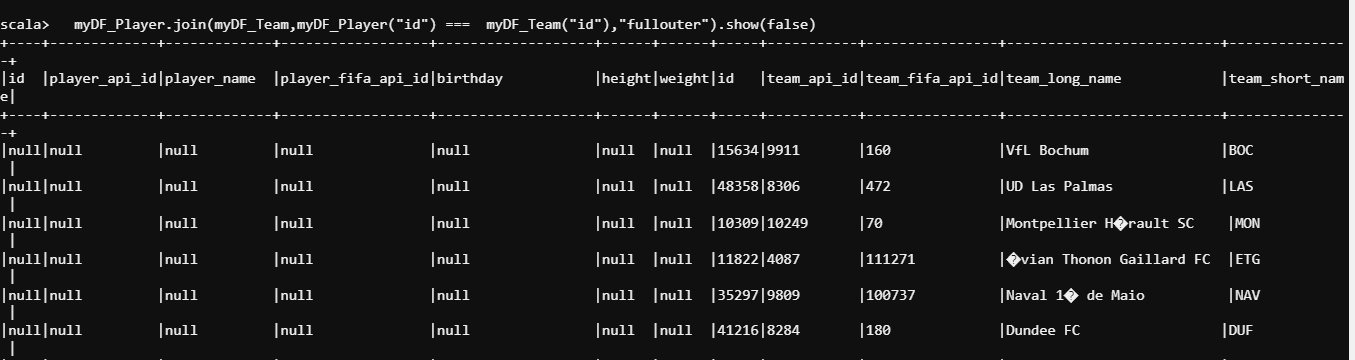
println("full join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"full").show(false)



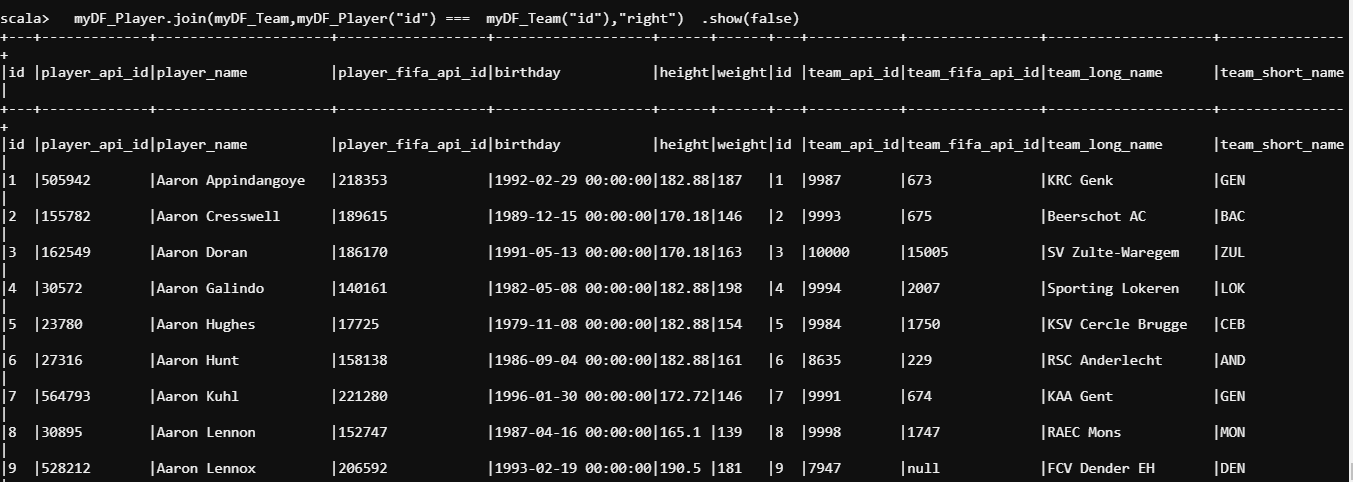
println("fullouter join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"fullouter") .show(false)



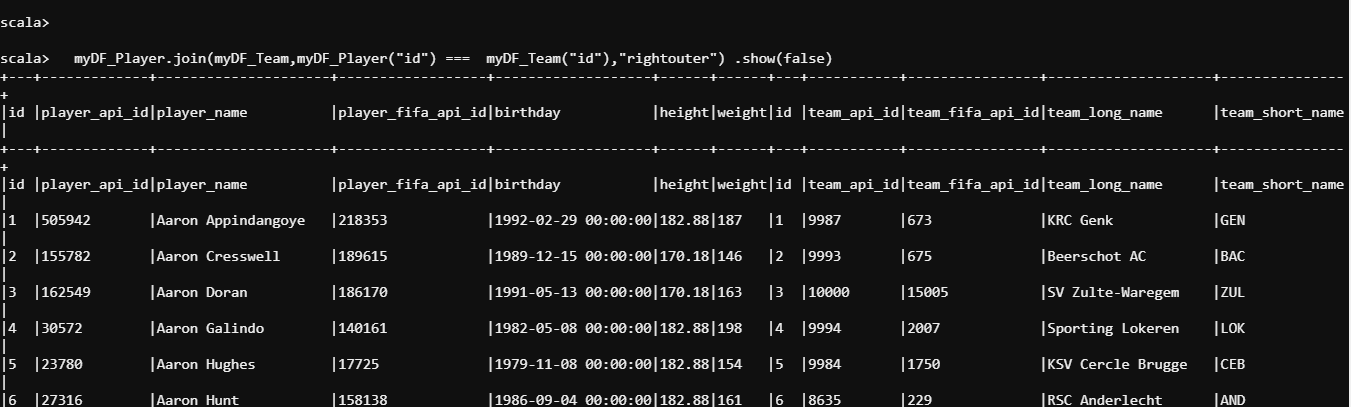
println("right join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"right") .show(false)



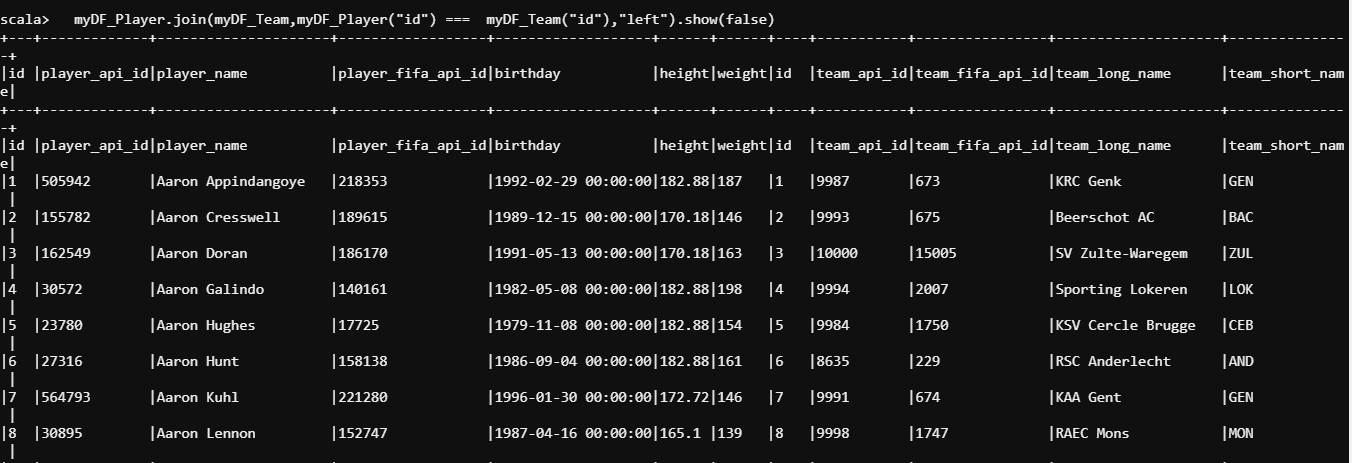
println("rightouter join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"rightouter") .show(false)



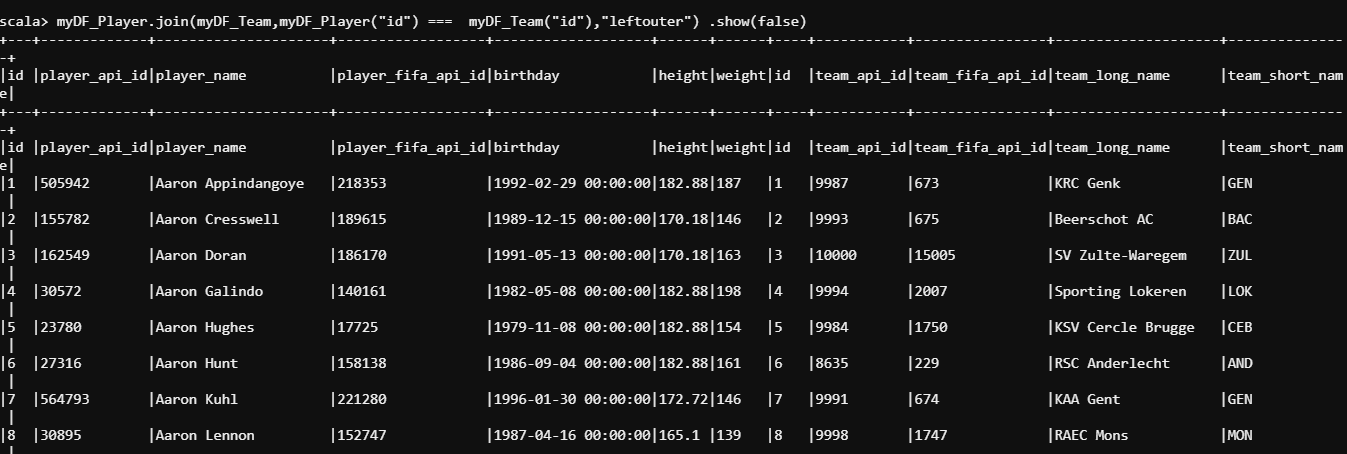
println("left join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"left").show(false)



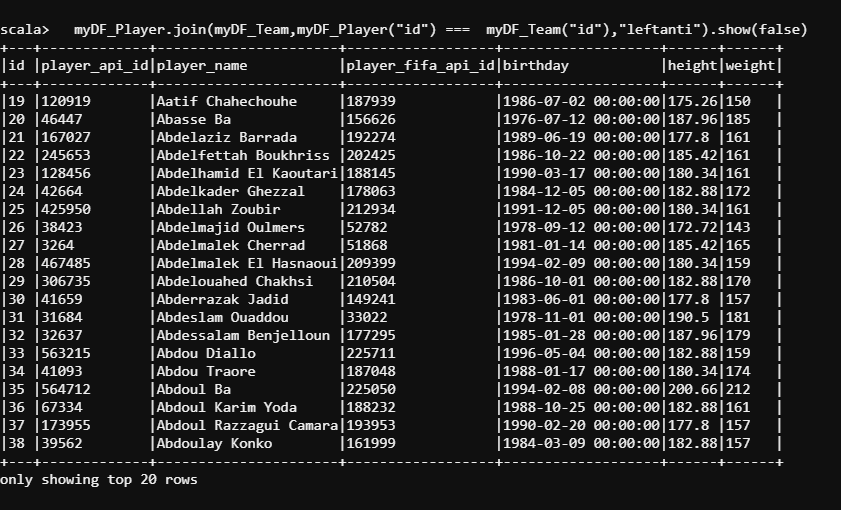
println("leftouter join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"leftouter") .show(false)



println("leftanti join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"leftanti").show(false)



println("leftsemi join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"leftsemi")

.show(false)

println("cross join")

myDF\_Player.join(myDF\_Team,myDF\_Player("id") === myDF\_Team("id"),"cross")

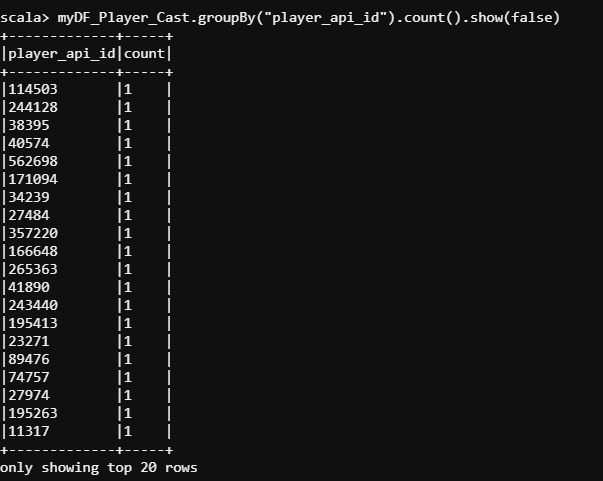
.show(false)

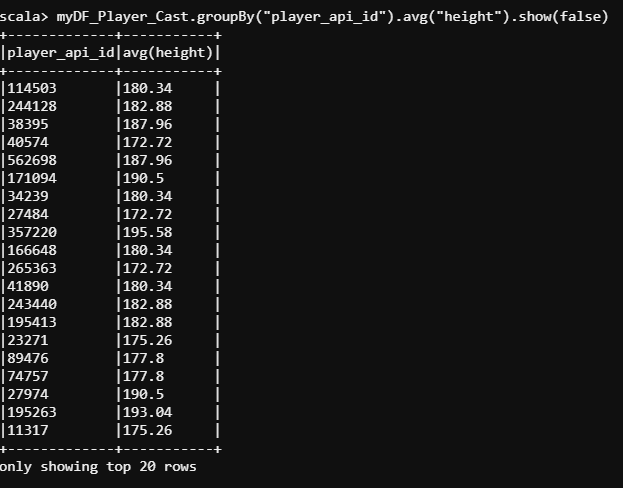
println("Using crossJoin()")

empDF.crossJoin(deptDF).show(false)

//Group By on single column

==============================

myDF\_Player\_Cast.groupBy("player\_api\_id").count().show(false)

myDF\_Player\_Cast.groupBy("player\_api\_id").avg("height").show(false)

myDF\_Player\_Cast.groupBy("player\_api\_id").sum("height").show(false)

myDF\_Player\_Cast.groupBy("player\_api\_id").max ("height").show(false)

myDF\_Player\_Cast.groupBy("player\_api\_id").min ("height").show(false)