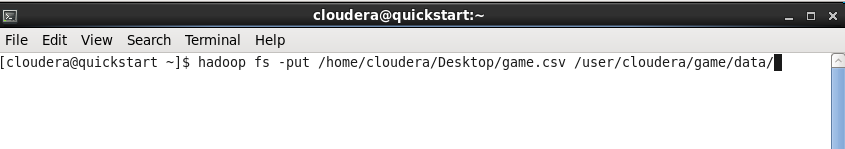
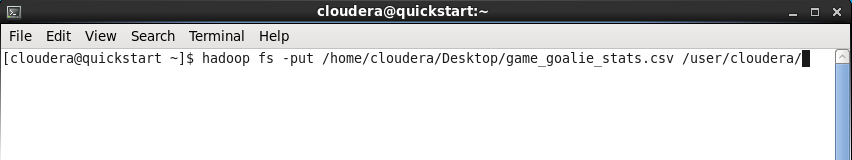
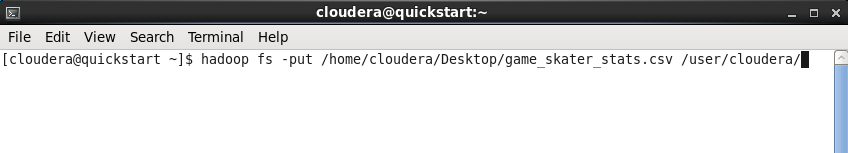
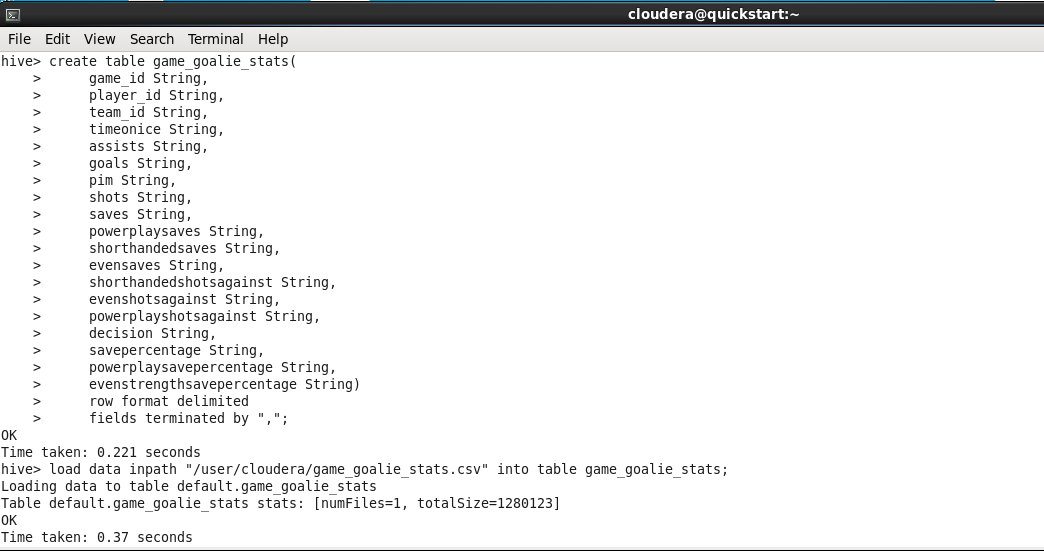
1. Loading all 3 csv files in HDFS.

* Load game.csv
* Load game\_goalie.csv
* Load game\_skater.csv



1. Creating tables in hive.

Data is loaded directly into hive using HiveQL.

Internal tables are created and data is loaded into them using ‘LOAD DATA INPATH’

**CLEANING**

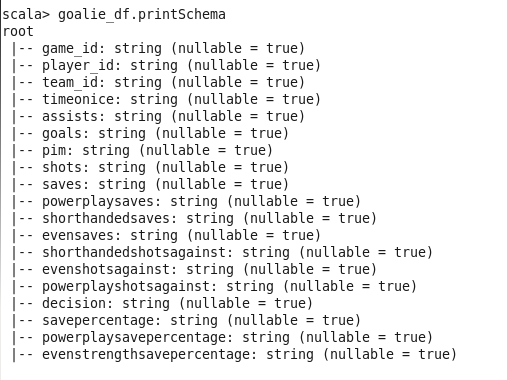
Dataframes are created from hive tables.

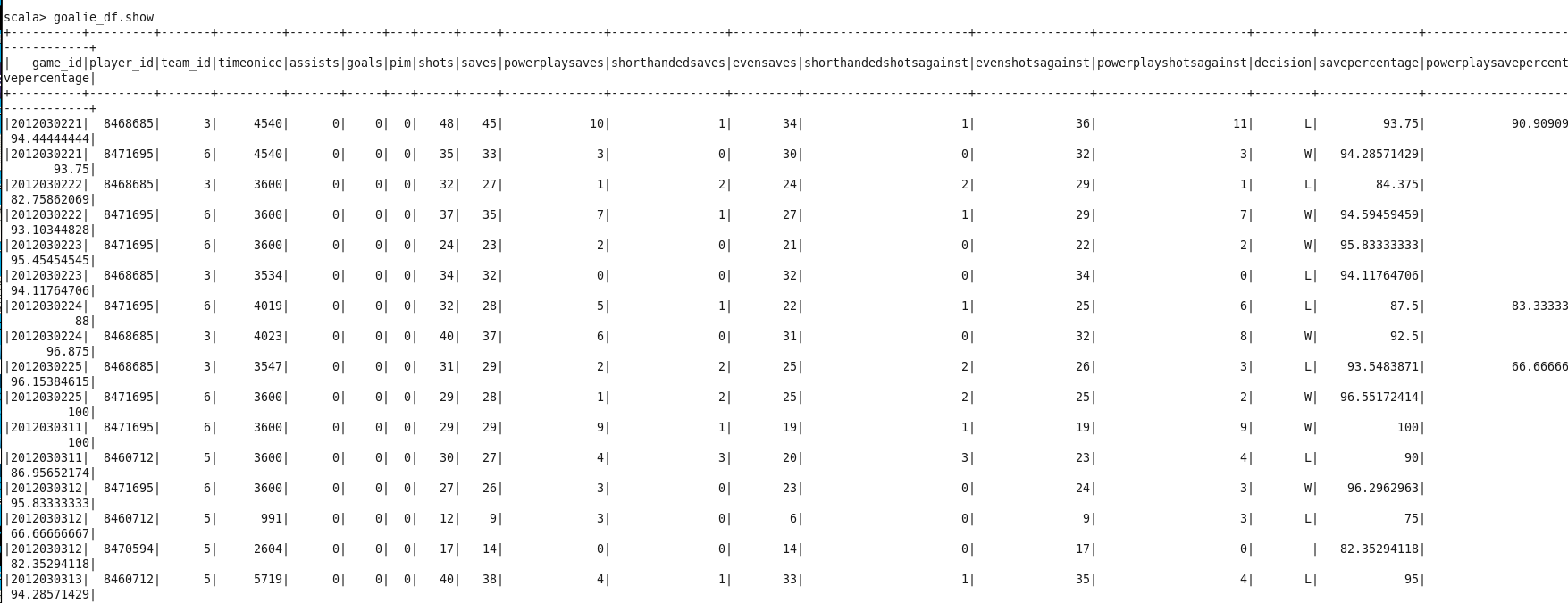
**CLEANING game\_goalie\_stats file**

Creating dataframe for game\_goalie\_stats

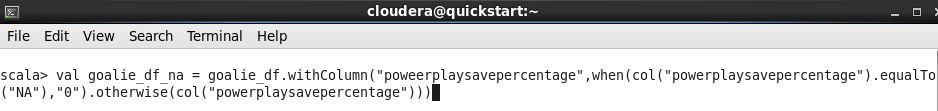
****

Game\_goalie\_stats Schema

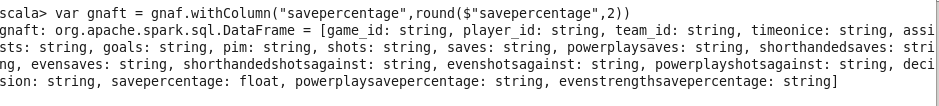


goalie\_df.show

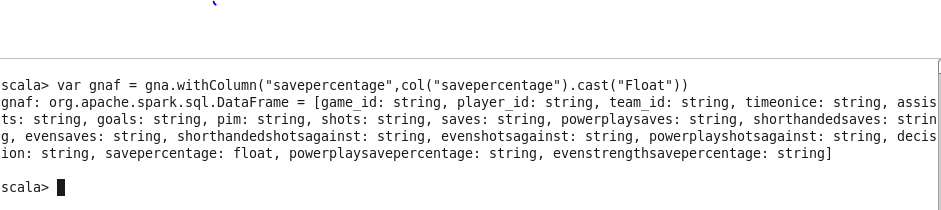
**Replacing NULL Values**



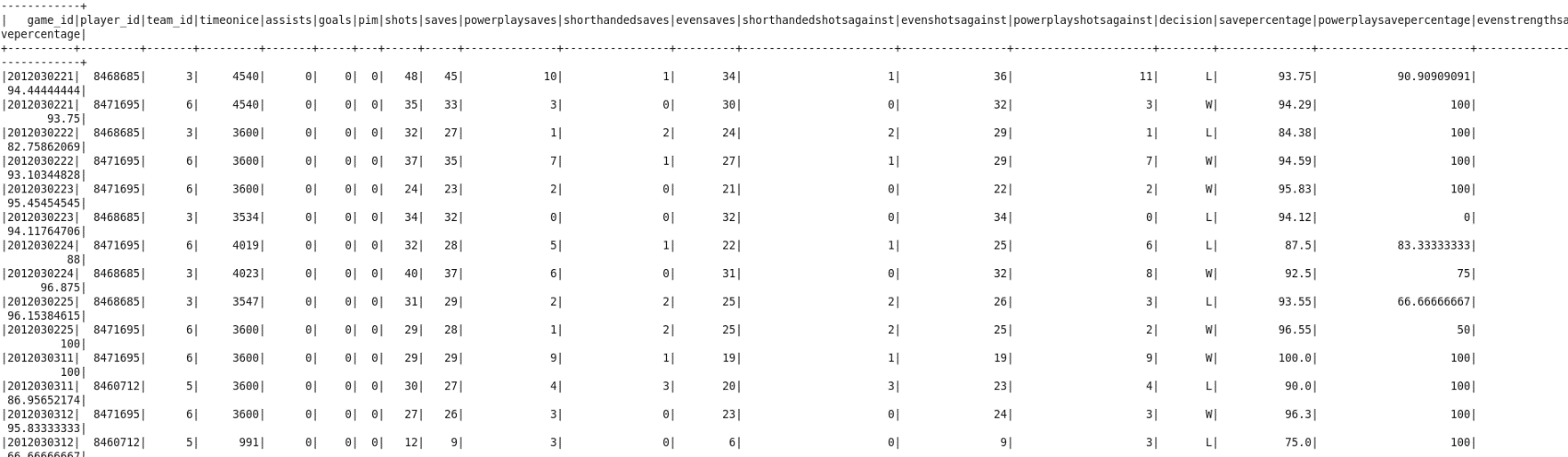
val goalie\_df\_na = goalie\_df.withColumn(“powerplaysavepercentage”,when(col(“powerplaysavepercentage”).equalTo(“NA”),”0”).otherwise(col(“powerplaysavepercentage”)))

**Round off to 2 decimal places**

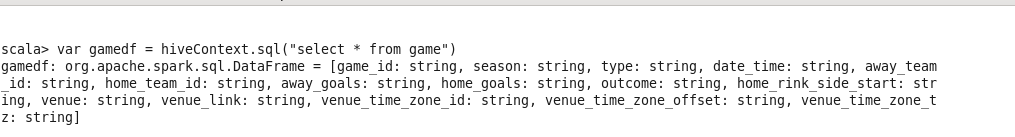
var gnaft = gnaf.withColumn(“savepercentage”,round($”savepercentage”,2))

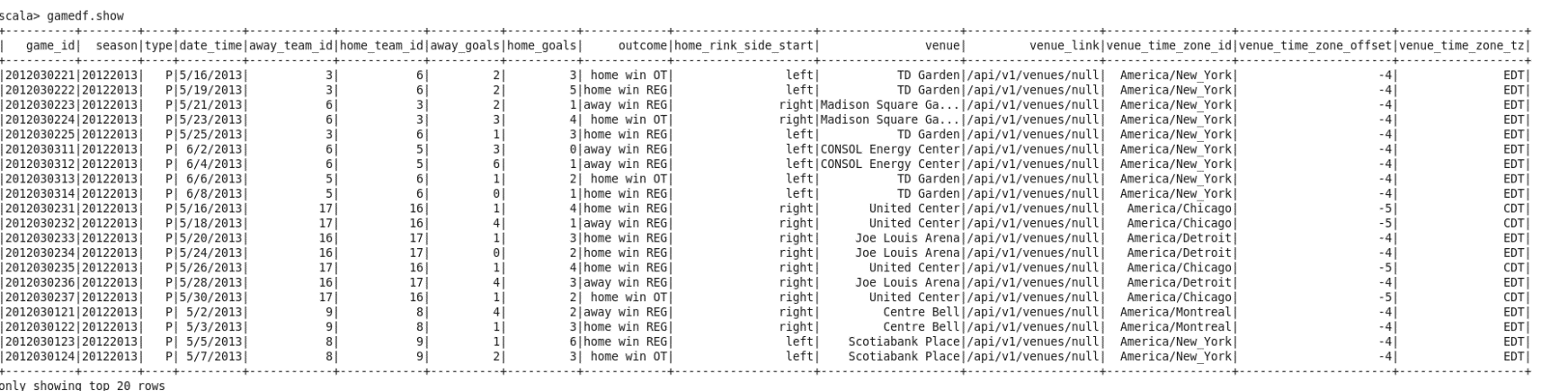
**convert Float to Integer**

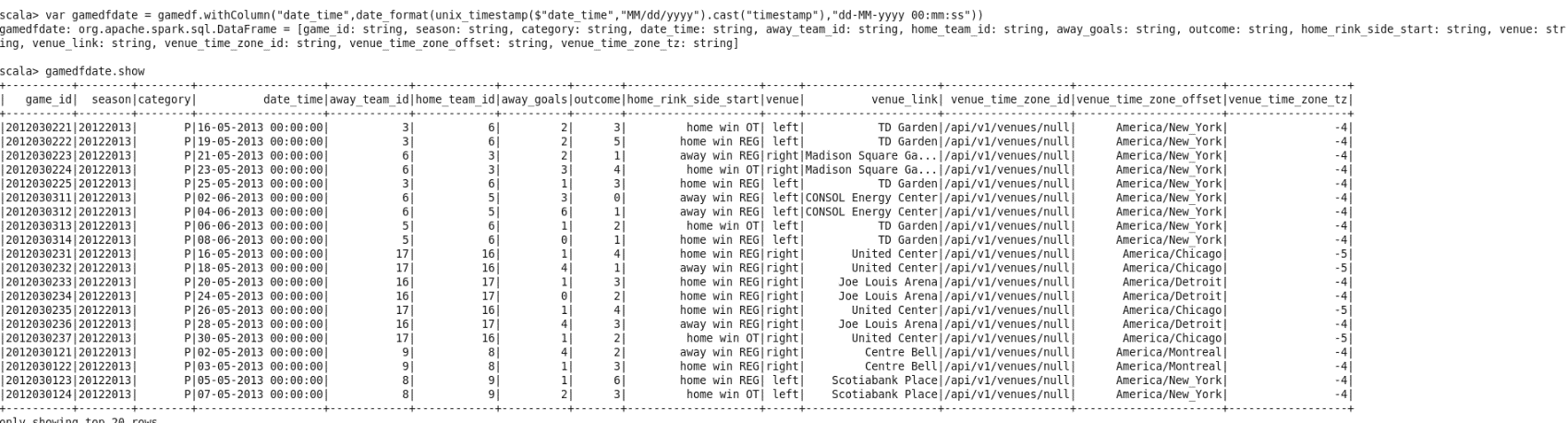
var gnaf = gna.withColumn(“savepercentage”,col(“savepercentage”).cast(‘Float”))

game\_goalie dataframe show

**CLEANING game.csv**

Creating dataframe for game.csv

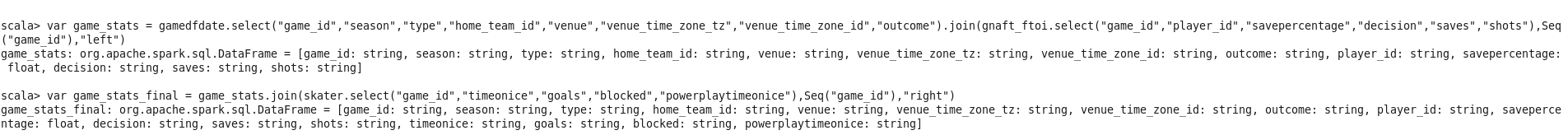
Dataframe

Converting date to dd-mm-yyyy hh:mm:ss

var gamedfdate = gamedf.withColumn(“date\_time”,date\_format(unix\_timestamp($”date\_time”,”MM/dd/yyyy”).cast(“timestamp”),”dd-MM-yyyy 00:mm:ss”))

**JOINING TABLES**

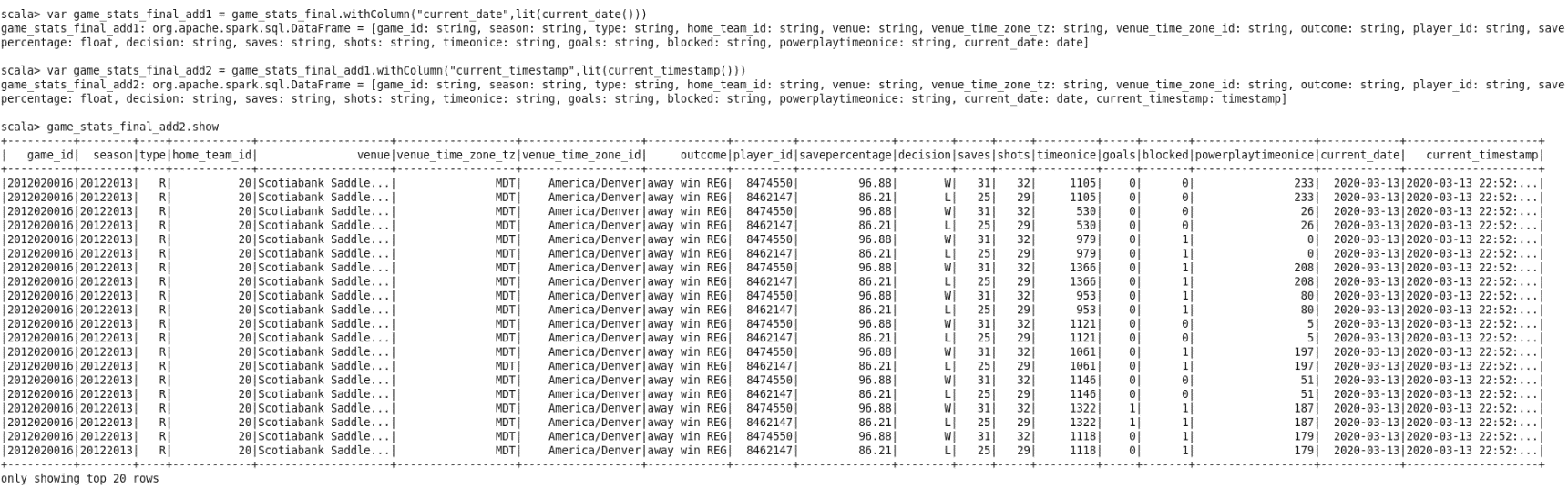
Dataframes are joined.

Joined dataframes: gamedfdate,gnaft\_ftoi,skater

var game\_stats = gamedfdate.select(“game\_id”,”season”,”type”,”home\_team\_id”,”venue”,\_time\_zone\_tz”,”venue\_time\_zone\_id”,”outcome”).join(gnaft\_ftoi.select(“game\_id”,”player\_id”,”saverpercentage”,”decision”,seq(“game\_id”),”left”)

var game\_stats\_final = game.stats.join(skater.select(“game\_id”,”timeonice”,”goals”,”blocked”,”powerplaytimeonice”,Seq(“game\_id”),”right”)

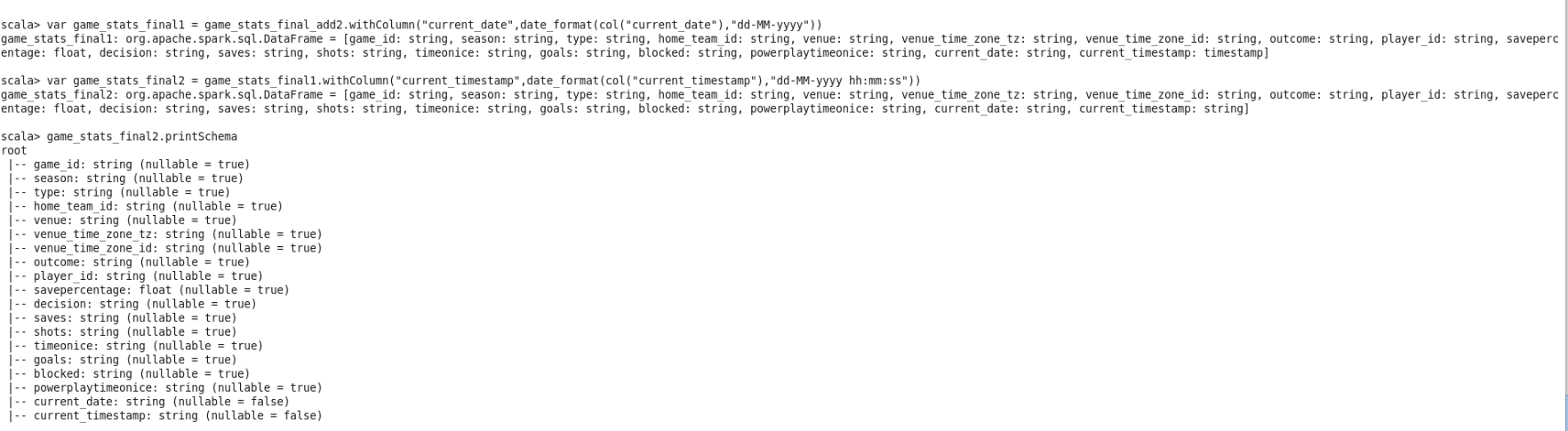
Adding current date and current timestamp to the joined table

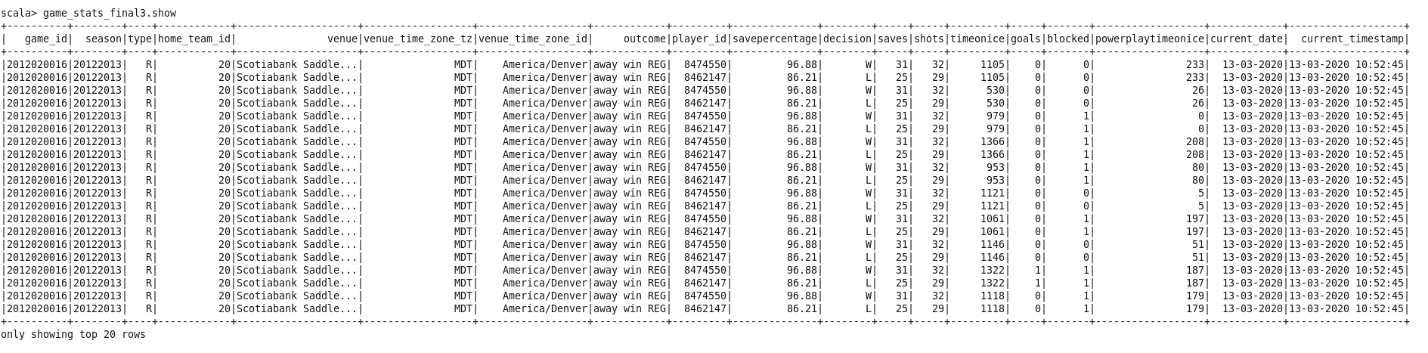


var game\_stats\_final\_add1 =

game\_stats\_final.withColumn(“current\_date,lit(current\_date()))

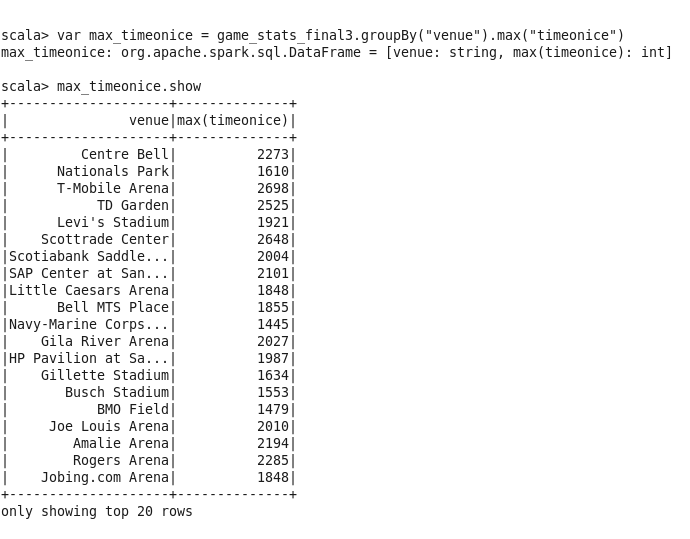
var game\_stats\_final\_add2 = game\_stats\_final\_add1.withColumn(“current\_timestamp”,lit(current\_timestamp()))

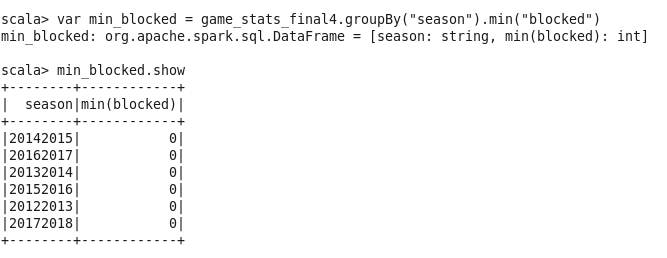
Converting current date and current timestamp columns to string

Final Table

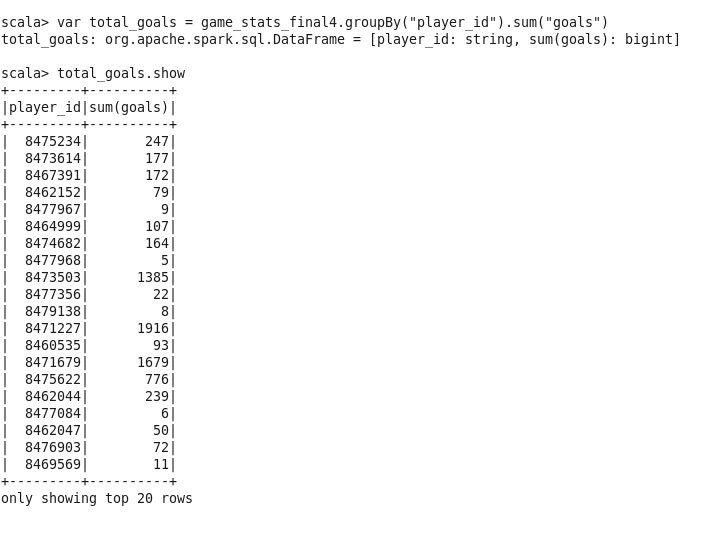
Loading data into hive

**AGGREGATION**

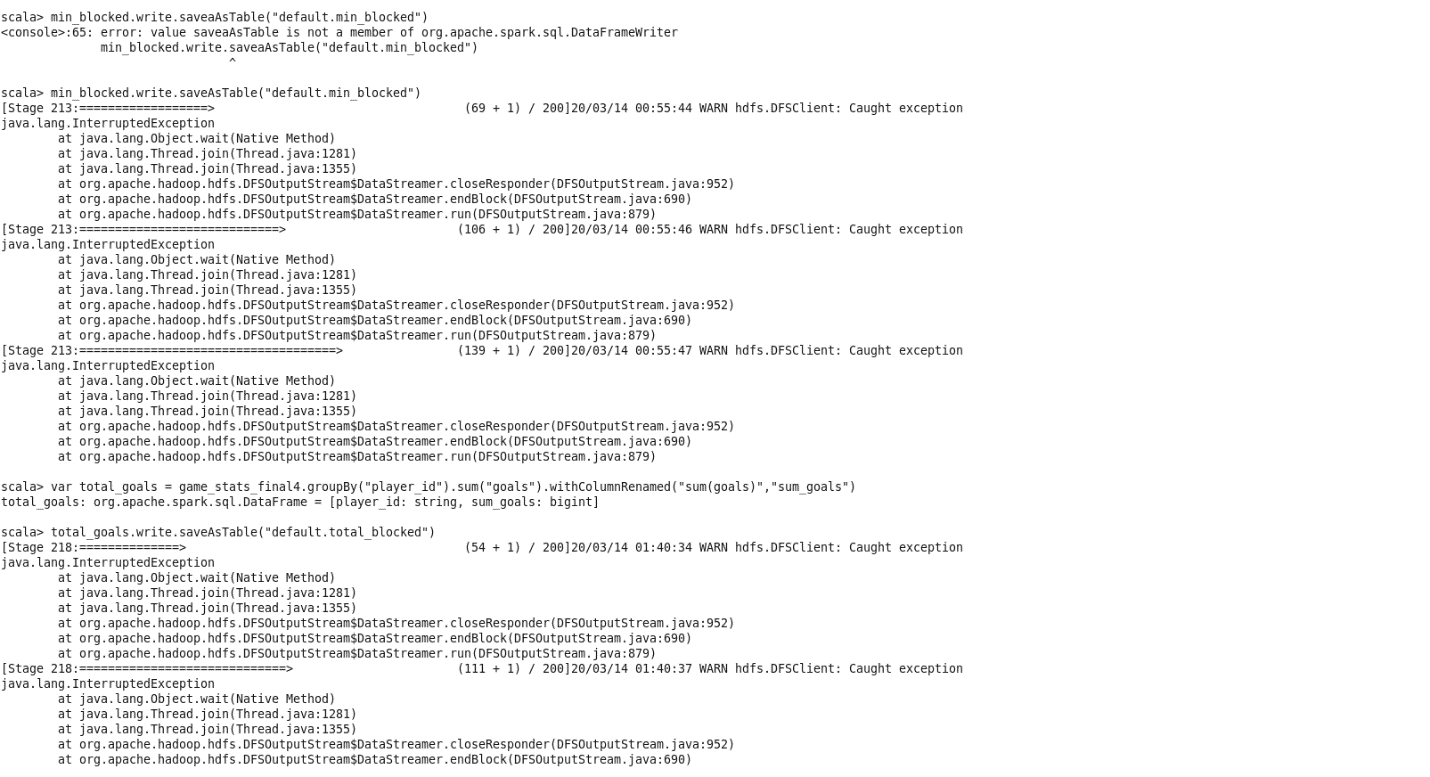
1. Find max of timeonice and group by venue
2. Find min blocked and group by season



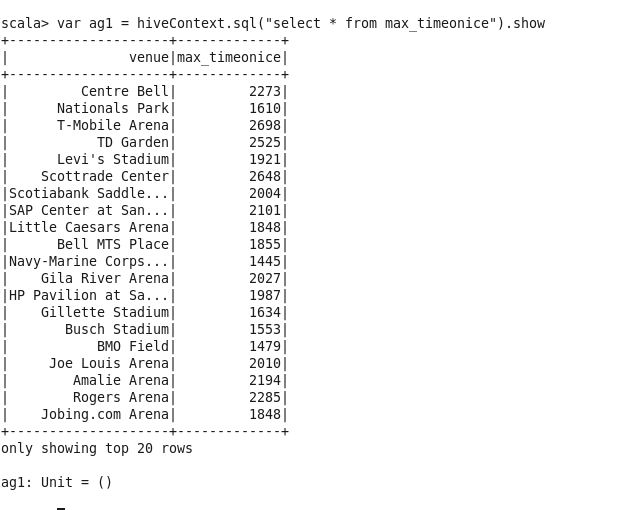
1. Find sum of goals group by player\_id



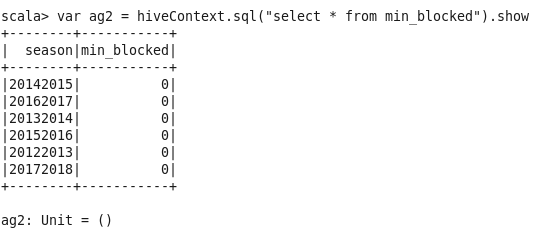
Saving aggregated dataframes to hive

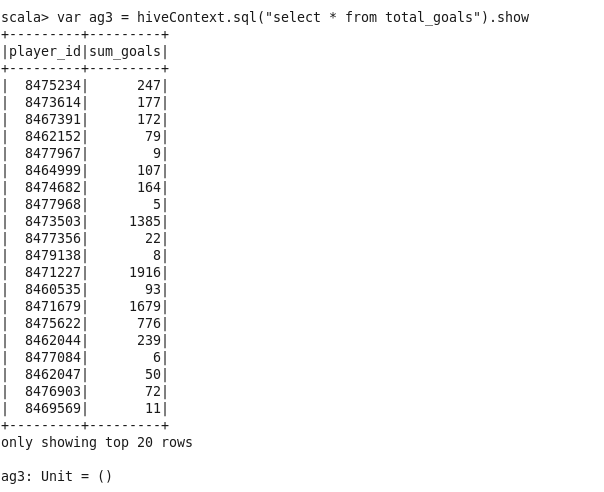


Aggregated Table 1

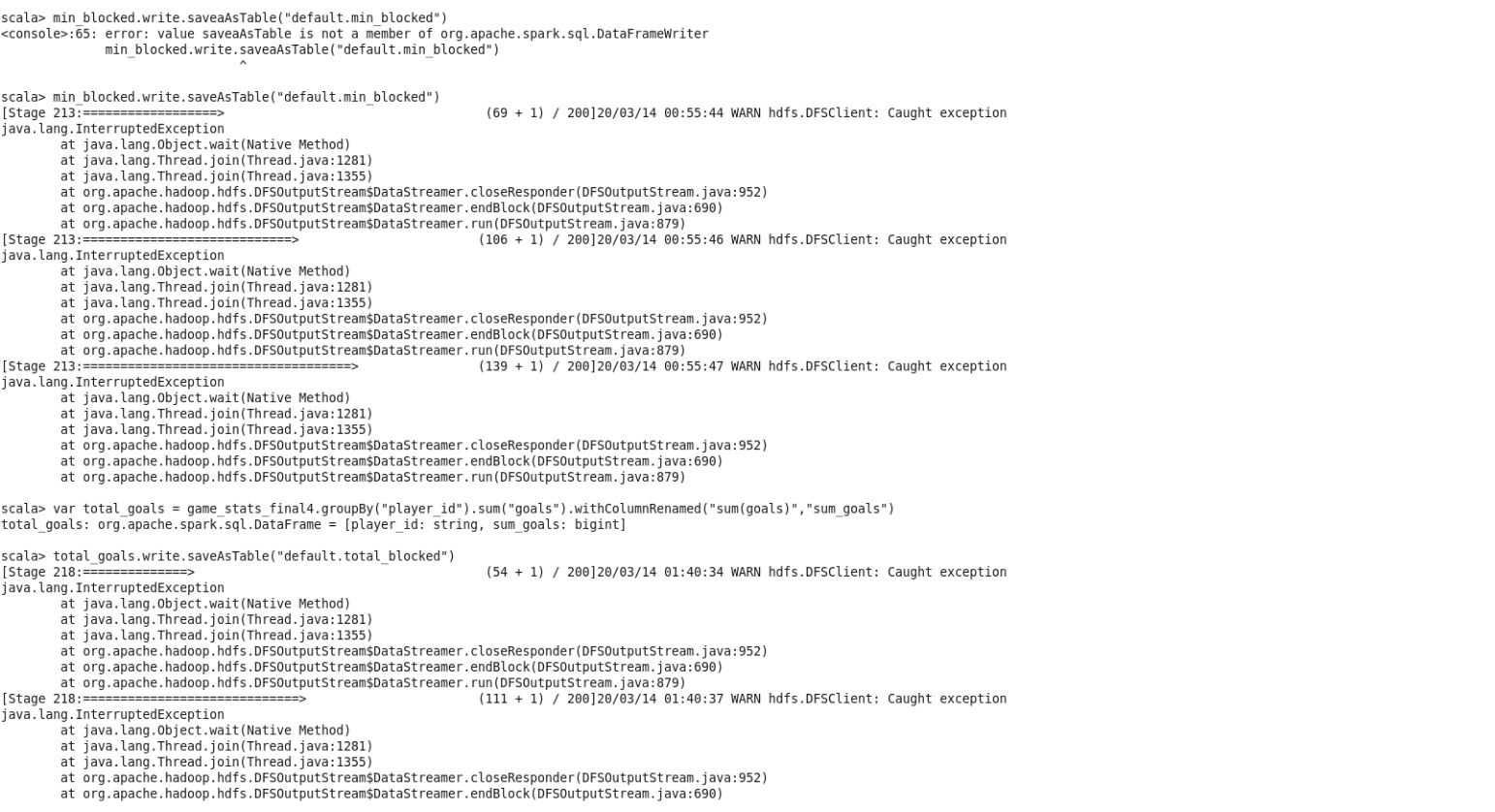


Aggregated Table 2



Aggregated Table 3

Saving Aggregated Tables into hive



**Final Tables in HIVE:**

**Original Tables:**

game

game\_stats

skater

**Cleansed Tables:**

gamefinal

game\_stats\_final

**Aggregated Tables:**

max\_timeonice

min\_blocked

total\_goals