Name: Arul Kumar ARK

Roll No.: 225229103

```
In [0]:
                                                                M
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.sql.types import Row
from datetime import datetime
In [0]:
spark = SparkSession.builder.appName("Python Spark SQL basic example").config("spark.son
In [0]:
student_records = sc.parallelize([Row(roll_no=1,name='John Doe',passed=True,marks={'Math
In [0]:
                                                                M
student_records_df = student_records.toDF()
student_records_df.show()
|roll_no|
           name passed
                               marks
                                              sports
DoB
1 John Doe true (Chemistry -> 81,... [chess, football] 2012
-05-01 13:01:05
     2|John Smith| false|{Chemistry -> 36,...|[volleyball, tabl...|2012
-05-12 14:02:05
```

```
In [0]:
                                                H
student records df.show(truncate=False)
+-----+
-----+
          |passed|marks
|roll_no|name
ports
             DoB
+-----+
     -----+
    |John Doe | true | {Chemistry -> 81, Math -> 89, Physics -> 87}|
[chess, football]
          |2012-05-01 13:01:05|
    |John Smith|false |{Chemistry -> 36, Math -> 29, Physics -> 31}|
[volleyball, tabletennis]|2012-05-12 14:02:05|
-----+
In [0]:
                                                M
student_records_df.createOrReplaceTempView('records')
In [0]:
                                                M
spark.sql("SELECT * FROM records").show()
|roll no|
        name|passed|
                       marks
                                   sports
DoB
1 John Doe true (Chemistry -> 81,... [chess, football] 2012
-05-01 13:01:05
   2|John Smith| false|{Chemistry -> 36,...|[volleyball, tabl...|2012
-05-12 14:02:05
In [0]:
                                                M
spark.sql('SELECT roll_no, marks["Physics"], sports[1] FROM records').show()
+----+
|roll no|marks[Physics]| sports[1]|
+----+
           87 football
   1|
   2
           31|tabletennis|
+----+
```

```
In [0]:
                                                 H
spark.sql("SELECT * FROM records where passed = True").show()
----+
       name|passed|
                       marks
|roll_no|
DoB
1|John Doe| true|{Chemistry -> 81,...|[chess, football]|2012-05-0
----+
In [0]:
                                                 H
spark.sql('SELECT * FROM records WHERE
     marks["Chemistry"] < 40').show()</pre>
sports|
       name|passed|
                        marks
|roll_no|
DoB
2|John Smith| false|{Chemistry -> 36,...|[volleyball, tabl...|2012
-05-12 14:02:05
----+
In [0]:
                                                 M
# Creating Global View
In [0]:
                                                 H
student_records_df.createGlobalTempView('global_record')
In [0]:
spark.sql("SELECT * FROM global_temp.global_records").show()
In [0]:
                                                 M
# Dropping Columns from DataFrame
```

```
In [0]:
                                                              H
student_records_df.columns
Out[19]: ['roll_no', 'name', 'passed', 'marks', 'sports', 'DoB']
In [0]:
                                                              M
student_records_df = student_records_df.drop('passed')
In [0]:
                                                              M
# Few More Queries
In [0]:
spark.sql("SELECT round( (marks.Physics+marks.Chemistry+marks.Math)/3) avg marks FROM re
+----+
avg_marks
+----+
    86.0
    32.0
+----+
                                                              M
In [0]:
student_records_df=spark.sql("SELECT *, round( (marks.Physics+marks.Chemistry+marks.Math
student_records_df.show()
----+
|roll no|
          name|passed|
                              marks
                                             sports|
DoB avg marks
-----+
     1 John Doe true (Chemistry -> 81,... [chess, football] 2012
-05-01 13:01:05
               86.0
     2|John Smith| false|{Chemistry -> 36,...|[volleyball, tabl...|2012
               32.0
-05-12 14:02:05
-----+
In [0]:
                                                              H
student_records_df.createOrReplaceTempView('records')
```

```
In [0]:
student_records_df = student_records_df.withColumn('status',(when(col('avg_marks')>=40,
student_records_df.show()
-----+
|roll no|
          name|passed|
                                marks
                                               sports
DoB avg_marks status
-----+
   1 John Doe true (Chemistry -> 81,... [chess, football] 2012
-05-01 13:01:05
               86.0 passed
     2|John Smith| false|{Chemistry -> 36,...|[volleyball, tabl...|2012
-05-12 14:02:05
              32.0|failed|
-----+
                                                                 M
In [0]:
# another table
In [0]:
                                                                 M
employeeData =(('John', 'HR', 'NY', 90000, 34, 10000),
('Neha','HR','NY',86000,28,20000),
('Robert', 'Sales', 'CA', 81000, 56, 22000),
('Maria', 'Sales', 'CA', 99000, 45, 15000),
('Paul','IT','NY',98000,38,14000),
('Jen','IT','CA',90000,34,20000),
('Raj','IT','CA',93000,28,28000),
('Pooja','IT','CA',95000,31,19000))
columns = ('employee_name', 'department', 'state', 'salary', 'age', 'bonus')
In [0]:
employeeDf = spark.createDataFrame(employeeData, columns)
In [0]:
employeeDf.groupby(col('department')).agg(sum(col('salary'))).show()
+----+
|department|sum(salary)|
+----+
      HR |
            176000
    Sales
            180000
      IT
            376000
```

```
In [0]:
employeeDf.groupby(col('department')).agg(sum(col('salary')).alias('total_sal')).orderBy
+----+
|department|total sal|
+-----+
       HR
            176000
    Sales|
            180000
       IT|
            376000
+-----+
In [0]:
                                                                     M
employeeDf.groupby(col('department')).agg(sum(col('salary')).alias('total_sal')).orderBy
+-----+
|department|total_sal|
+-----+
       IT
          376000
    Sales
            180000
       HR
            176000
 ------
In [0]:
                                                                     M
employeeDf.groupby(col('department'),col('state')).agg(sum(col('bonus'))).show()
+-----+
|department|state|sum(bonus)|
+-----+
          NY
                  30000
       HR |
    Sales
            CA
                  37000
            NY
                  14000
       IT|
            CA|
       IT|
                  67000
 -----+
In [0]:
                                                                     M
employeeDf.groupby(col('department')).agg(avg(col('salary')).alias('avarage_salary'),max
+----+
|department|avarage_salary|maximum_bonus|
                            20000
       HR |
               88000.0
    Sales
               90000.0
                            22000
       IT
               94000.0
                            28000
```

In [0]:	M
In [0]:	H
In [0]:	H