qsn

February 10, 2026

```
[1]: from pyspark import SparkContext

     sc=SparkContext(master="local",appName="test1")

     df=sc.textFile("students.txt")

     df.collect()
```

```
[1]: ['8955Tarun,Male,78:50:45:25',
      '8871Ali,Male,100:100:99:98',
      '8892Rafi,Male,100:91:100:99',
      '8912Uday,Male,100:97:99:98',
      '8910Random,Female,80:40:60:54']
```

```
[ ]: df_split=df.map(lambda x:x.split(","))
     df_split.collect()
```

```
[ ]: [['8955Tarun', 'Male', '78:50:45:25'],
      ['8871Ali', 'Male', '100:100:99:98'],
      ['8892Rafi', 'Male', '100:91:100:99'],
      ['8912Uday', 'Male', '100:97:99:98'],
      ['8910Random', 'Female', '80:40:60:54']]
```

```
[ ]: df_formatted=df_split.map(lambda x:(x[0][:4],x[0][4:],x[1],*x[2].split(":")))
     df_formatted.collect()
```

```
[ ]: [('8955', 'Tarun', 'Male', '78', '50', '45', '25'),
      ('8871', 'Ali', 'Male', '100', '100', '99', '98'),
      ('8892', 'Rafi', 'Male', '100', '91', '100', '99'),
      ('8912', 'Uday', 'Male', '100', '97', '99', '98'),
      ('8910', 'Random', 'Female', '80', '40', '60', '54')]
```

```
[ ]: # total Marks
     total_marks=df_formatted.map(lambda x:(int(x[3])+int(x[4])+int(x[5])+int(x[6]))/
      ↪4)
     total_marks.collect()
```

```
Grade=total_marks.map(grade)
Grade.collect()
```

```
[ ]: ['C', 'A', 'A', 'A', 'C']
```

```
[25]: def grade(avg):
          if avg>=80:
              return "A"
          elif avg>=60:
              return "B"
          elif avg>=40:
              return "C"
          else:
              return "F"
      df_final=df_formatted.map(lambda x:x+((int(x[3]) + int(x[4]) + int(x[5]) +␣
        ↪int(x[6])) ,)).map(lambda x:x+(grade(x[7]/4),))

      df_final.collect()
```

```
[25]: [('8955', 'Tarun', 'Male', '78', '50', '45', '25', 198, 'C'),
       ('8871', 'Ali', 'Male', '100', '100', '99', '98', 397, 'A'),
       ('8892', 'Rafi', 'Male', '100', '91', '100', '99', 390, 'A'),
       ('8912', 'Uday', 'Male', '100', '97', '99', '98', 394, 'A'),
       ('8910', 'Random', 'Female', '80', '40', '60', '54', 234, 'C')]
```

```
[30]: df_final.filter(lambda x:x[8]=="A" or x[8]=="B").collect()
```

```
[30]: [('8871', 'Ali', 'Male', '100', '100', '99', '98', 397, 'A'),
       ('8892', 'Rafi', 'Male', '100', '91', '100', '99', 390, 'A'),
       ('8912', 'Uday', 'Male', '100', '97', '99', '98', 394, 'A')]
```

```
[42]: header =␣
        ↪("number","name","gender","maths","physics","biology","chemistry","total_marks","grade")

      header_rdd=sc.parallelize([header])

      res=header_rdd.union(df_final)
      # res.collect()
      for row in res.collect():
          print(",".join(map(str,row)))
```

```
number,name,gender,maths,physics,biology,chemistry,total_marks,grade
8955,Tarun,Male,78,50,45,25,198,C
8871,Ali,Male,100,100,99,98,397,A
8892,Rafi,Male,100,91,100,99,390,A
8912,Uday,Male,100,97,99,98,394,A
8910,Random,Female,80,40,60,54,234,C
```