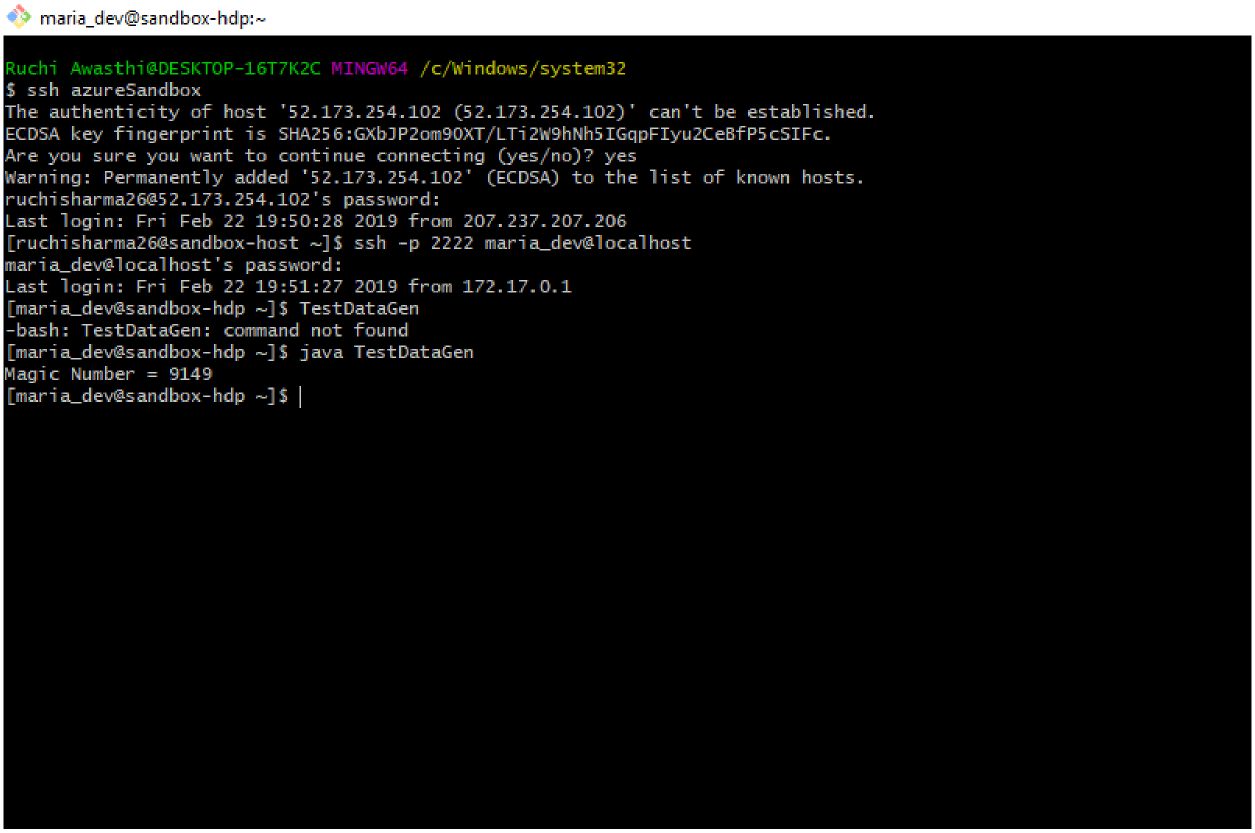
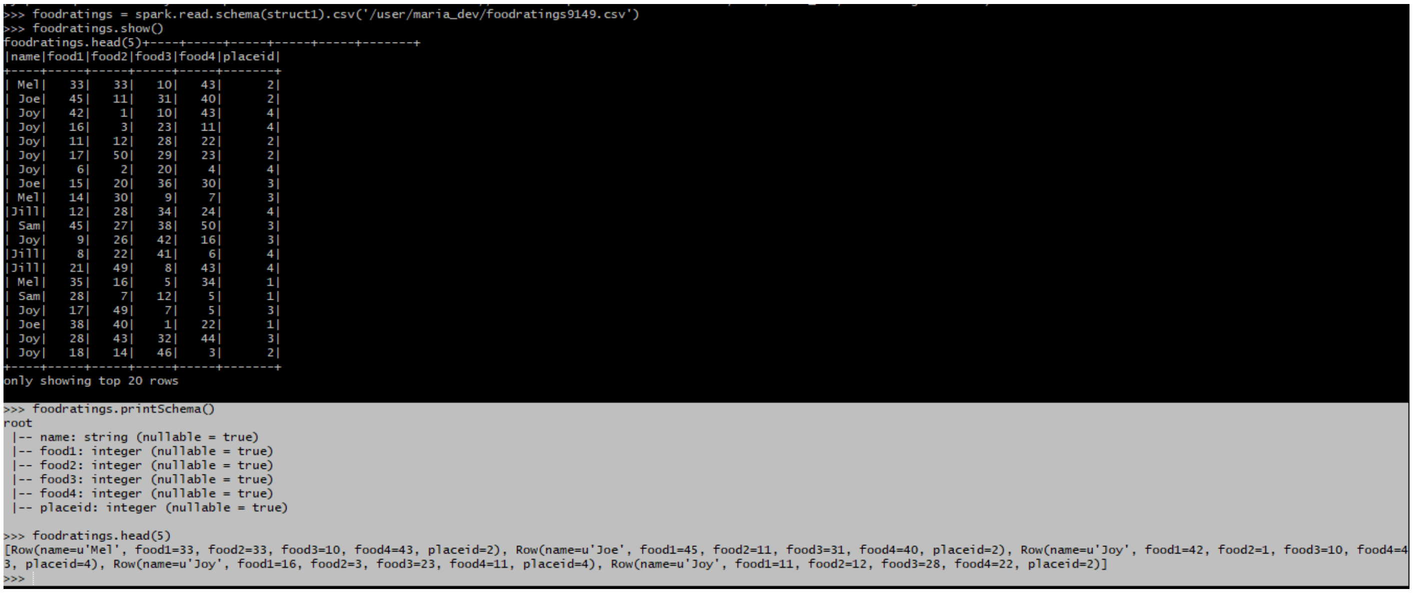
Magic Number: 9149



```
Ruchi Awasthi@DESKTOP-16T7K2C MINGW64 /c/Windows/system32
$ ssh azureSandbox
The authenticity of host '52.173.254.102 (52.173.254.102)' can't be established.
ECDSA key fingerprint is SHA256:GXbJP2om90XT/LTi2W9hNh5IGqpFIyu2CeBfP5cSIFc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.173.254.102' (ECDSA) to the list of known hosts.
ruchisharma26@52.173.254.102's password:
Last login: Fri Feb 22 19:50:28 2019 from 207.237.207.206
[ruchisharma26@sandbox-host ~]$ ssh -p 2222 maria_dev@localhost
maria_dev@localhost's password:
Last login: Fri Feb 22 19:51:27 2019 from 172.17.0.1
[maria_dev@sandbox-hdp ~]$ TestDataGen
-bash: TestDataGen: command not found
[maria_dev@sandbox-hdp ~]$ java TestDataGen
Magic Number = 9149
[maria_dev@sandbox-hdp ~]$ |
```

Exercise 1) Magic Number: 9149



```
>>> foodratings = spark.read.schema(struct1).csv('/user/maria_dev/foodratings9149.csv')
>>> foodratings.show()
foodratings.head(5)+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|   33|   33|   10|   43|      2|
| Joe|   45|   11|   31|   40|      2|
| Joy|   42|    1|   10|   43|      4|
| Joy|   16|    3|   23|   11|      4|
| Joy|   11|   12|   28|   22|      2|
| Joy|   17|   50|   29|   23|      2|
| Joy|    6|    2|   20|    4|      4|
| Joe|   15|   20|   36|   30|      3|
| Mel|   14|   30|    9|    7|      3|
|Jill|   12|   28|   34|   24|      4|
| Sam|   45|   27|   38|   50|      3|
| Joy|    9|   26|   42|   16|      3|
|Jill|    8|   22|   41|    6|      4|
|Jill|   21|   49|    8|   43|      4|
| Mel|   35|   16|    5|   34|      1|
| Sam|   28|    7|   12|    5|      1|
| Joy|   17|   49|    7|    5|      3|
| Joe|   38|   40|    1|   22|      1|
| Joy|   28|   43|   32|   44|      3|
| Joy|   18|   14|   46|    3|      2|
+----+-----+-----+-----+-----+-------+
only showing top 20 rows

>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Joe', food1=45, food2=11, food3=31, food4=40, placeid=2), Row(name=u'Joy', food1=42, food2=1, food3=10, food4=4
3, placeid=4), Row(name=u'Joy', food1=16, food2=3, food3=23, food4=11, placeid=4), Row(name=u'Joy', food1=11, food2=12, food3=28, food4=22, placeid=2)]
>>>
```

```
hadoop fs -copyFromLocal /home/maria_dev/foodratings9149.txt /user/maria_dev/foodratings9149.csv
hadoop fs -copyFromLocal /home/maria_dev/foodplaces9149.txt /user/maria_dev/foodplaces9149.csv
```

```
from pyspak.sql.types import *
struct1 = StructType(
[
StructField("name", StringType(), True),
StructField("food1",IntegerType(), True),
StructField("food2",IntegerType(), True),
StructField("food3",IntegerType(), True),
StructField("food4",IntegerType(), True),
StructField("placeid",IntegerType(), True)
]
)
foodratings = spark.read.schema(struct1).csv('/user/maria_dev/foodratings9149.csv')
foodratings.printSchema()
foodratings.head(5)
```

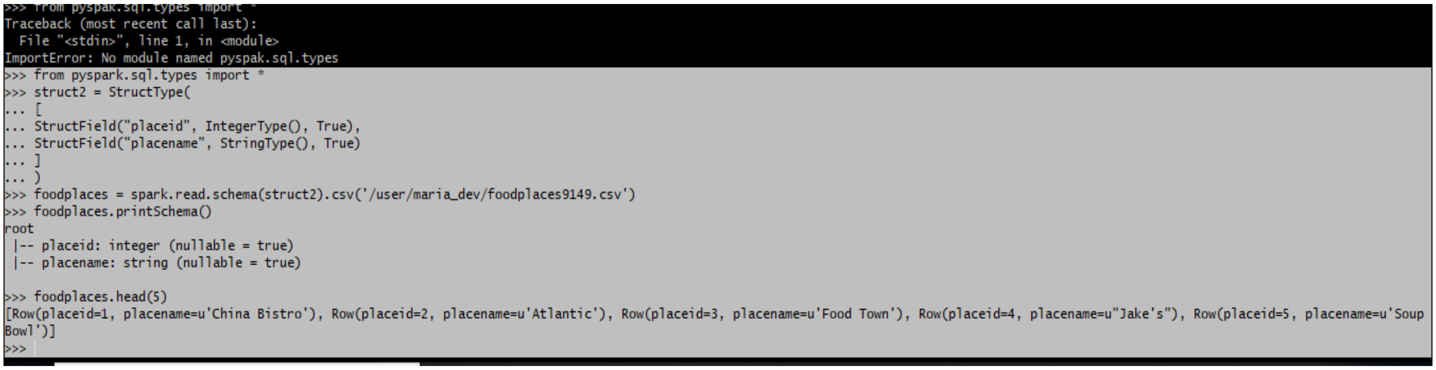
Exercise 2)

```
struct2 = StructType(
[
StructField("placeid", IntegerType(), True),
StructField("placename", StringType(), True)
]
)
```

foodplaces = spark.read.schema(struct2).csv('/user/maria_dev/foodplaces9149.csv')

foodplaces.printSchema()

foodplaces.head(5)



```
>>> from pyspak.sql.types import *
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named pyspak.sql.types
>>> from pyspark.sql.types import *
>>> struct2 = StructType(
... [
... StructField("placeid", IntegerType(), True),
... StructField("placename", StringType(), True)
... ]
... )
>>> foodplaces = spark.read.schema(struct2).csv('/user/maria_dev/foodplaces9149.csv')
>>> foodplaces.printSchema()
root
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

>>> foodplaces.head(5)
[Row(placeid=1, placename=u'China Bistro'), Row(placeid=2, placename=u'Atlantic'), Row(placeid=3, placename=u'Food Town'), Row(placeid=4, placename=u"Jake's"), Row(placeid=5, placename=u'Soup
Bowl')]
>>>
```

<u>Exercise 3)</u>

foodratings.registerTempTable('foodratingsT')

foodratings_ex3=sqlContext.sql("SELECT * FROM foodratingsT WHERE food2<25 AND food4>40")

foodratings_ex3.head(5)

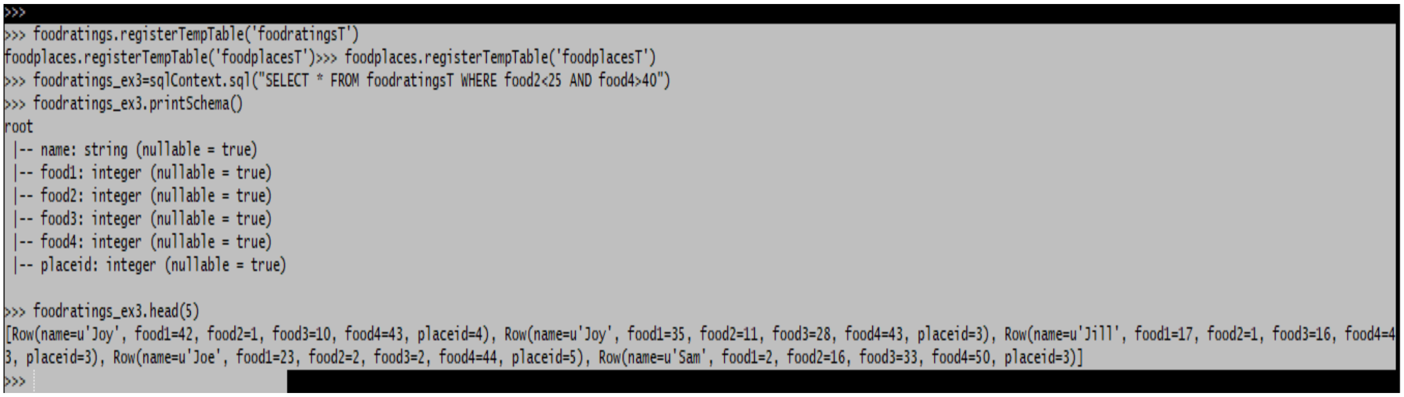
foodratings_ex3.printSchema()

foodplaces.registerTempTable('foodplacesT')

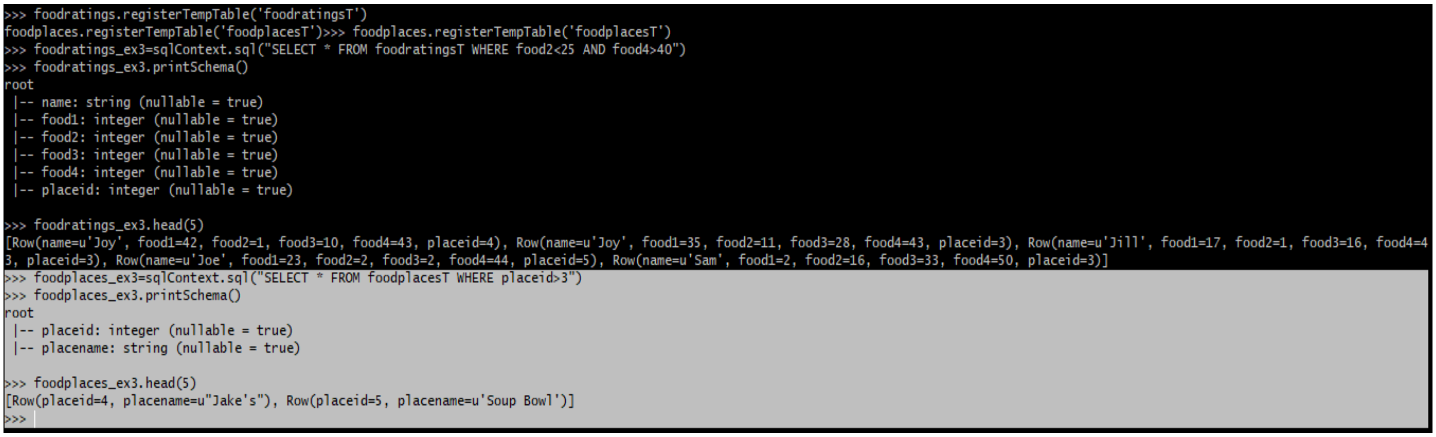
foodplaces_ex3=sqlContext.sql("SELECT * FROM foodplacesT WHERE placeid>3")

foodplaces_ex3.printSchema()

foodplaces_ex3.head(5)

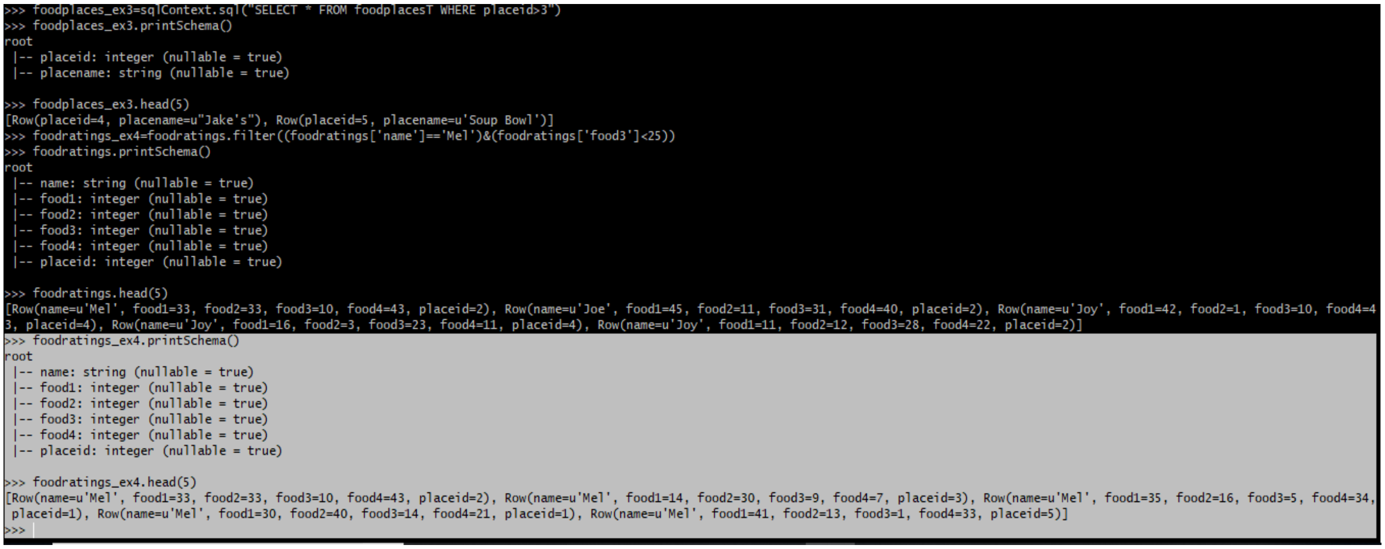


```
>>>
>>> foodratings.registerTempTable('foodratingsT')
foodplaces.registerTempTable('foodplacesT')>>> foodplaces.registerTempTable('foodplacesT')
>>> foodratings_ex3=sqlContext.sql("SELECT * FROM foodratingsT WHERE food2<25 AND food4>40")
>>> foodratings_ex3.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex3.head(5)
[Row(name=u'Joy', food1=42, food2=1, food3=10, food4=43, placeid=4), Row(name=u'Joy', food1=35, food2=11, food3=28, food4=43, placeid=3), Row(name=u'Jill', food1=17, food2=1, food3=16, food4=4
3, placeid=3), Row(name=u'Joe', food1=23, food2=2, food3=2, food4=44, placeid=5), Row(name=u'Sam', food1=2, food2=16, food3=33, food4=50, placeid=3)]
>>>
```



```
>>> foodratings.registerTempTable('foodratingsT')
foodplaces.registerTempTable('foodplacesT')>>> foodplaces.registerTempTable('foodplacesT')
>>> foodratings_ex3=sqlContext.sql("SELECT * FROM foodratingsT WHERE food2<25 AND food4>40")
>>> foodratings_ex3.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex3.head(5)
[Row(name=u'Joy', food1=42, food2=1, food3=10, food4=43, placeid=4), Row(name=u'Joy', food1=35, food2=11, food3=28, food4=43, placeid=3), Row(name=u'Jill', food1=17, food2=1, food3=16, food4=4
3, placeid=3), Row(name=u'Joe', food1=23, food2=2, food3=2, food4=44, placeid=5), Row(name=u'Sam', food1=2, food2=16, food3=33, food4=50, placeid=3)]
>>> foodplaces_ex3=sqlContext.sql("SELECT * FROM foodplacesT WHERE placeid>3")
>>> foodplaces_ex3.printSchema()
root
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

>>> foodplaces_ex3.head(5)
[Row(placeid=4, placename=u"Jake's"), Row(placeid=5, placename=u'Soup Bowl')]
>>>
```

Exercise 4)



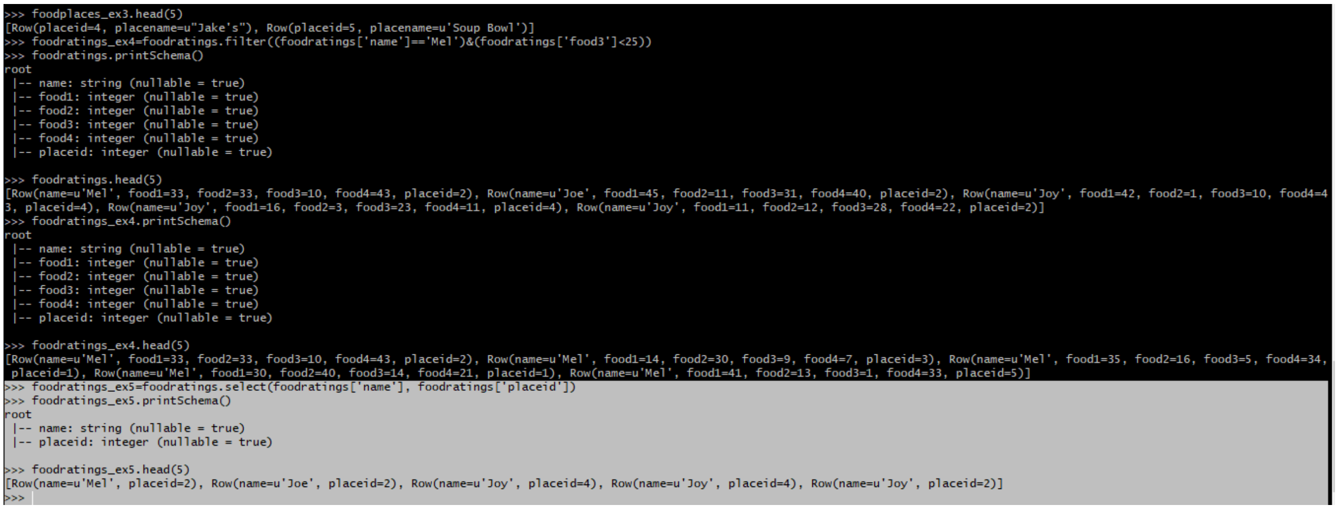
```
>>> foodplaces_ex3=sqlContext.sql("SELECT * FROM foodplacesT WHERE placeid>3")
>>> foodplaces_ex3.printSchema()
root
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

>>> foodplaces_ex3.head(5)
[Row(placeid=4, placename=u"Jake's"), Row(placeid=5, placename=u'Soup Bowl')]
>>> foodratings_ex4=foodratings.filter((foodratings['name']=='Mel')&(foodratings['food3']<25))
>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Joe', food1=45, food2=11, food3=31, food4=40, placeid=2), Row(name=u'Joy', food1=42, food2=1, food3=10, food4=4
3, placeid=4), Row(name=u'Joy', food1=16, food2=3, food3=23, food4=11, placeid=4), Row(name=u'Joy', food1=11, food2=12, food3=28, food4=22, placeid=2)]
>>> foodratings_ex4.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex4.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Mel', food1=14, food2=30, food3=9, food4=7, placeid=3), Row(name=u'Mel', food1=35, food2=16, food3=5, food4=34,
 placeid=1), Row(name=u'Mel', food1=30, food2=40, food3=14, food4=21, placeid=1), Row(name=u'Mel', food1=41, food2=13, food3=1, food4=33, placeid=5)]
>>>
```

foodratings_ex5=foodratings.filter((foodratings['name']=='Mel')&(foodratings['food3']<25))

foodratings_ex4.printSchema()

foodratings_ex4.head(5)

Exercise 5)



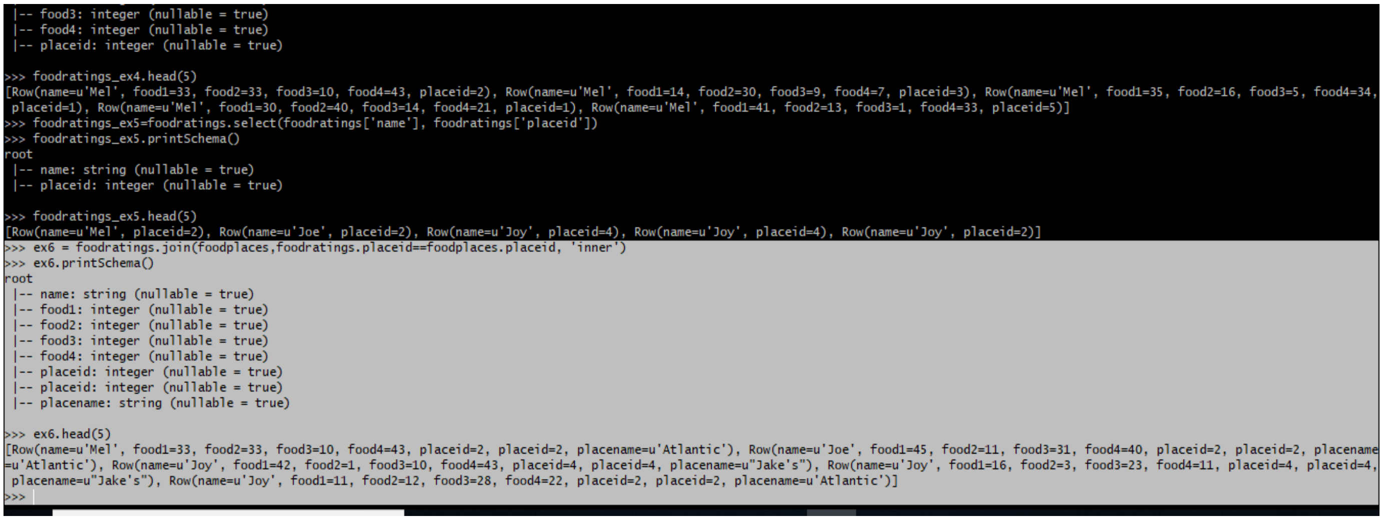
```
>>> foodplaces_ex3.head(5)
[Row(placeid=4, placename=u"Jake's"), Row(placeid=5, placename=u'Soup Bowl')]
>>> foodratings_ex4=foodratings.filter((foodratings['name']=='Mel')&(foodratings['food3']<25))
>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Joe', food1=45, food2=11, food3=31, food4=40, placeid=2), Row(name=u'Joy', food1=42, food2=1, food3=10, food4=4
3, placeid=4), Row(name=u'Joy', food1=16, food2=3, food3=23, food4=11, placeid=4), Row(name=u'Joy', food1=11, food2=12, food3=28, food4=22, placeid=2)]
>>> foodratings_ex4.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex4.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Mel', food1=14, food2=30, food3=9, food4=7, placeid=3), Row(name=u'Mel', food1=35, food2=16, food3=5, food4=34,
 placeid=1), Row(name=u'Mel', food1=30, food2=40, food3=14, food4=21, placeid=1), Row(name=u'Mel', food1=41, food2=13, food3=1, food4=33, placeid=5)]
>>> foodratings_ex5=foodratings.select(foodratings['name'], foodratings['placeid'])
>>> foodratings_ex5.printSchema()
root
 |-- name: string (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex5.head(5)
[Row(name=u'Mel', placeid=2), Row(name=u'Joe', placeid=2), Row(name=u'Joy', placeid=4), Row(name=u'Joy', placeid=4), Row(name=u'Joy', placeid=2)]
>>>
```

foodratings_ex5=foodratings.select(foodratings['name'], foodratings['placeid'])

foodratings_ex5.printSchema()

foodratings_ex5.head(5)

Exercise 6)



```
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex4.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2), Row(name=u'Mel', food1=14, food2=30, food3=9, food4=7, placeid=3), Row(name=u'Mel', food1=35, food2=16, food3=5, food4=34,
 placeid=1), Row(name=u'Mel', food1=30, food2=40, food3=14, food4=21, placeid=1), Row(name=u'Mel', food1=41, food2=13, food3=1, food4=33, placeid=5)]
>>> foodratings_ex5=foodratings.select(foodratings['name'], foodratings['placeid'])
>>> foodratings_ex5.printSchema()
root
 |-- name: string (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex5.head(5)
[Row(name=u'Mel', placeid=2), Row(name=u'Joe', placeid=2), Row(name=u'Joy', placeid=4), Row(name=u'Joy', placeid=4), Row(name=u'Joy', placeid=2)]
>>> ex6 = foodratings.join(foodplaces,foodratings.placeid==foodplaces.placeid, 'inner')
>>> ex6.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

>>> ex6.head(5)
[Row(name=u'Mel', food1=33, food2=33, food3=10, food4=43, placeid=2, placeid=2, placename=u'Atlantic'), Row(name=u'Joe', food1=45, food2=11, food3=31, food4=40, placeid=2, placeid=2, placename
=u'Atlantic'), Row(name=u'Joy', food1=42, food2=1, food3=10, food4=43, placeid=4, placeid=4, placename=u"Jake's"), Row(name=u'Joy', food1=16, food2=3, food3=23, food4=11, placeid=4, placeid=4,
 placename=u"Jake's"), Row(name=u'Joy', food1=11, food2=12, food3=28, food4=22, placeid=2, placeid=2, placename=u'Atlantic')]
>>>
```

```
ex6 = foodratings.join(foodplaces,foodratings.placeid==foodplaces.placeid, 'inner')

ex6.printSchema()

ex6.head(5)
```