seek_interview

March 18, 2023

[1]:
```
## Download Test Data
!wget -q https://coding-challenge-public.s3.ap-southeast-2.amazonaws.com/
 ↪test-data.zip
```

[2]:
```
## Unzip Test Data
! unzip  -o  -P By9FNTZXp4j4izuufAs= ~/test-data.zip -d ~/
```

```
Archive:  /home/jovyan/test-data.zip
  inflating: /home/jovyan/test_data/part18.json
  inflating: /home/jovyan/test_data/part8.json
  inflating: /home/jovyan/test_data/part14.json
  inflating: /home/jovyan/test_data/part4.json
  inflating: /home/jovyan/test_data/part5.json
  inflating: /home/jovyan/test_data/part15.json
  inflating: /home/jovyan/test_data/part9.json
  inflating: /home/jovyan/test_data/part19.json
  inflating: /home/jovyan/test_data/part12.json
  inflating: /home/jovyan/test_data/part2.json
  inflating: /home/jovyan/test_data/part3.json
  inflating: /home/jovyan/test_data/part13.json
  inflating: /home/jovyan/test_data/part0.json
  inflating: /home/jovyan/test_data/part10.json
  inflating: /home/jovyan/test_data/part11.json
  inflating: /home/jovyan/test_data/part1.json
  inflating: /home/jovyan/test_data/part6.json
  inflating: /home/jovyan/test_data/part16.json
  inflating: /home/jovyan/test_data/part17.json
  inflating: /home/jovyan/test_data/part7.json
```

[3]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").appName("seek_interview").
 ↪getOrCreate()
```

[4]:
```
spark
```

[4]:
```
<pyspark.sql.session.SparkSession at 0x7f118af5f310>
```

[6]:
```
# 1. Load the dataset into a Spark dataframe.
# 2. Print the schema
# 3. How many records are there in the dataset?

test_data_df = spark.read.json("test_data/*.json")
number_of_rows = test_data_df.count()
test_data_df.printSchema()
print("There's %d rows in Test Data"%number_of_rows)
```

```
root
 |-- id: string (nullable = true)
 |-- profile: struct (nullable = true)
 |    |-- firstName: string (nullable = true)
 |    |-- jobHistory: array (nullable = true)
 |    |    |-- element: struct (containsNull = true)
 |    |    |    |-- fromDate: string (nullable = true)
 |    |    |    |-- location: string (nullable = true)
 |    |    |    |-- salary: long (nullable = true)
 |    |    |    |-- title: string (nullable = true)
 |    |    |    |-- toDate: string (nullable = true)
 |    |-- lastName: string (nullable = true)

There's 17139693 rows in Test Data
```

[7]:
```
# 4. What is the average salary for each profile? Display the first 10 results, ordered by lastName in descending order.
test_data_df.createOrReplaceTempView("test_data_tab")
#q4=spark.sql("select id,profile.firstName,profile.lastName,profile.jobHistory from test_data_tab limit 5").show(5)
q1=spark.sql(
            "Select id, \
                    firstName, \
                    lastName, \
                    avg(jobHistory.salary) salary \
            from \
                (select id, \
                    profile.firstName,\
                    profile.lastName, \
                    explode(profile.jobHistory) as jobHistory \
                 from test_data_tab \
                 ) \
            group by 1,2,3 \
            order by lastName desc \
            limit 10"
            ).show()
```

```
+--------------------+---------+--------+------------------+
```

```
|                  id|firstName|lastName|            salary|
+--------------------+---------+--------+------------------+
|82dab74c-3946-45b…|   Robert|  Zywiec| 66833.33333333333|
|ba24222d-6e39-40d…|  Matthew|  Zywiec|           65500.0|
|5894afab-574f-429…|  Richard|  Zywiec|           69625.0|
|8137bbb1-e6d6-4cb…|    Scott| Zywicki|           82500.0|
|cf56af73-988b-4b0…|   Joseph| Zywicki|           53625.0|
|e568d991-18c4-43c…|    Doris| Zywicki| 95666.66666666667|
|03aeca24-7be1-42a…|  Charles| Zywicki|           95000.0|
|40fa57e1-5f0e-45e…|    James| Zywicki|           86000.0|
|af1598d7-9faf-4cd…|  Therese| Zywicki|113166.66666666667|
|cc529ff4-2dbf-4ce…|  Cherryl| Zywicki|47666.666666666664|
+--------------------+---------+--------+------------------+
```

[8]:
```
#5. What is the average salary across the whole dataset?
q5=spark.sql(
            "Select avg(jobHistory.salary) salary \
            from \
                (select id, \
                    profile.firstName,\
                    profile.lastName, \
                    explode(profile.jobHistory) as jobHistory \
                 from test_data_tab \
                 ) \
            "
            ).show()
```

```
+----------------+
|          salary|
+----------------+
|97473.6229416272|
+----------------+
```

[9]:
```
#6. On average, what are the top 5 paying jobs? Bottom 5 paying jobs? If there␣
 ↪is a tie, please order by title, location.
q6_1=spark.sql(
            "Select title, \
                    salary, \
                    location, \
                    'High Paying jobs' Category, \
                    row_number() over (order by (salary)  desc , title asc,␣
 ↪location asc) rnk \
            from (\
                  Select distinct jobHistory.title, \
                        jobHistory.salary, \
```

```
                        jobHistory.location \
            from (\
                    select id, \
                        profile.firstName,\
                        profile.lastName, \
                        explode(profile.jobHistory) as jobHistory \
                     from test_data_tab \
                ) \
               ) \
            order by rnk asc \
            limit 5"
            ).show(5,False)

q6_2=spark.sql(
            "Select title, \
                    salary, \
                    location, \
                    'Low Paying jobs' Category, \
                    row_number() over (order by (salary)  asc , title asc,␣
 ↪location asc) rnk \
            from (\
                  Select distinct jobHistory.title, \
                        jobHistory.salary, \
                        jobHistory.location \
            from (\
                    select id, \
                        profile.firstName,\
                        profile.lastName, \
                        explode(profile.jobHistory) as jobHistory \
                     from test_data_tab \
                ) \
               ) \
            order by rnk asc \
            limit 5"
            ).show(5,False)
```

```
+----------------------+------+---------+----------------+---+
|title                 |salary|location |Category        |rnk|
+----------------------+------+---------+----------------+---+
|Administration Officer|159000|Adelaide |High Paying jobs|1  |
|Administration Officer|159000|Brisbane |High Paying jobs|2  |
|Administration Officer|159000|Canberra |High Paying jobs|3  |
|Administration Officer|159000|Hobart   |High Paying jobs|4  |
|Administration Officer|159000|Melbourne|High Paying jobs|5  |
+----------------------+------+---------+----------------+---+

+----------------------+------+---------+---------------+---+
```

```
|title                   |salary|location |Category       |rnk|
+------------------------+------+---------+---------------+---+
|counter manager         |-6000 |Melbourne|Low Paying jobs|1  |
|human resources manager|-6000 |Melbourne|Low Paying jobs|2  |
|registration officer    |-4000 |Canberra |Low Paying jobs|3  |
|Warehouse Storeperson   |-3000 |Melbourne|Low Paying jobs|4  |
|dental assistant        |-3000 |Hobart   |Low Paying jobs|5  |
+------------------------+------+---------+---------------+---+
```

[10]:
```
#7. Who is currently making the most money? If there is a tie, please order in␣
 ↪lastName descending, fromDate descending.
q7=spark.sql(
            "Select firstName, \
                    lastName, \
                    jobHistory.title, \
                    jobHistory.fromDate, \
                    max(jobHistory.salary) avg_salary, \
                    row_number() over ( order by max(jobHistory.salary)  desc ,␣
 ↪lastName desc, jobHistory.fromDate desc) rnk \
            from \
                (select id, \
                    profile.firstName,\
                    profile.lastName, \
                    explode(profile.jobHistory) as jobHistory \
                 from test_data_tab \
                 ) \
            group by 1,2,3,4 \
            order by rnk asc \
            limit 1"
            ).show(1,False)
```

```
+---------+---------+----------------------+----------+----------+---+
|firstName|lastName |title                 |fromDate  |avg_salary|rnk|
+---------+---------+----------------------+----------+----------+---+
|Sandra   |Zyskowski|procurement specialist|2015-04-11|159000    |1  |
+---------+---------+----------------------+----------+----------+---+
```

[11]:
```
#8. What was the most popular job title that started in 2019?
q8=spark.sql(
            "Select jobHistory.title, \
                    row_number() over ( order by count(distinct id)  desc) rnk \
            from \
                (select id, \
                    profile.firstName,\
                    profile.lastName, \
```

```
                    explode(profile.jobHistory) as jobHistory \
                from test_data_tab \
                ) \
            where year(jobHistory.fromDate)=2019 \
            group by 1 \
            order by rnk asc \
            limit 1"
            ).show(1,False)
```

```
+-----------------+---+
|title            |rnk|
+-----------------+---+
|Sheetmetal Worker|1  |
+-----------------+---+
```

[12]:
```
#9. How many people are currently working?
q9=spark.sql(
            "Select count(distinct id) current_workers_count \
            from \
                (select id, \
                    profile.firstName,\
                    profile.lastName, \
                    explode(profile.jobHistory) as jobHistory \
                from test_data_tab \
                ) \
            where jobHistory.toDate is null\
            limit 1"
            ).show(1,False)
```

```
+---------------------+
|current_workers_count|
+---------------------+
|7710613              |
+---------------------+
```

[13]:
```
#10. For each person, list only their latest job. Display the first 10 results,␣
 ↪ordered by lastName descending, firstName ascending order.
q10=spark.sql(
            "Select firstName, \
                    lastName, \
                    title \
            from( Select id, \
                    firstName, \
                    lastName, \
                    jobHistory.title, \
```

```
                    row_number() over (partition by id order by␣
 ↪coalesce(jobHistory.toDate,'9999') desc) rnk \
                from \
                    (select id, \
                        profile.firstName,\
                        profile.lastName, \
                        explode(profile.jobHistory) as jobHistory \
                     from test_data_tab \
                     ) \
                ) \
            where rnk=1 \
            order by lastName desc, firstName asc \
            limit 10"
            ).show(10,False)
```

```
+---------+--------+---------------------------+
|firstName|lastName|title                      |
+---------+--------+---------------------------+
|Matthew  |Zywiec  |Multi Site Manager         |
|Richard  |Zywiec  |assembler                  |
|Robert   |Zywiec  |registration officer       |
|Bobby    |Zywicki |taxation accountant        |
|Calvin   |Zywicki |assistant operations manager|
|Charles  |Zywicki |sales consultant           |
|Cherryl  |Zywicki |trimmer                    |
|Christine|Zywicki |internal sales             |
|Darlene  |Zywicki |evaluator                  |
|Donna    |Zywicki |internal sales             |
+---------+--------+---------------------------+
```

[14]:
```
#11. For each person, list their highest paying job along with their first␣
 ↪name, last name, salary and the year they made this salary.
#    Store the results in a dataframe, and then print out 10 results

q11=spark.sql(
            "Select firstName, \
                    lastName, \
                    salary ,\
                    high_sal_job_year \
            from( Select id, \
                    firstName, \
                    lastName, \
                    jobHistory.salary, \
                    year(jobHistory.fromDate) high_sal_job_year, \
                    row_number() over (partition by id order by jobHistory.
 ↪salary desc) rnk \
```

```
                from \
                    (select id, \
                        profile.firstName,\
                        profile.lastName, \
                        explode(profile.jobHistory) as jobHistory \
                     from test_data_tab \
                     ) \
                ) \
            where rnk=1 \
            order by lastName desc, firstName asc \
            "
            )
q11.show(10,False)
```

```
+---------+--------+------+-----------------+
|firstName|lastName|salary|high_sal_job_year|
+---------+--------+------+-----------------+
|Matthew  |Zywiec  |67000 |2017             |
|Richard  |Zywiec  |83000 |2018             |
|Robert   |Zywiec  |85000 |2016             |
|Bobby    |Zywicki |89000 |2017             |
|Calvin   |Zywicki |144000|2015             |
|Charles  |Zywicki |95000 |2016             |
|Cherryl  |Zywicki |66000 |2017             |
|Christine|Zywicki |71000 |2018             |
|Darlene  |Zywicki |76000 |2014             |
|Donna    |Zywicki |115000|2019             |
+---------+--------+------+-----------------+
only showing top 10 rows
```

[15]:
```
#12. Write out the last result (question 11) in parquet format, compressed,␣
 ↪partitioned by the year of their highest paying job.
q11.write.partitionBy("high_sal_job_year").mode('overwrite').
 ↪option("compression","gzip").parquet("q11.parquet")
```

[ ]:
```
```