

Nama: Ryan Bertrand

NIM: 2501967096

hr_db & gp_db Databases are available in Verulam Blue

```
scala> spark.sql("SHOW DATABASES").show()
+-----+
|databaseName|
+-----+
| bogus_db   |
| default    |
| emr_db     |
| gp_db      |
| hr_db      |
| sales_db   |
| solutions_db |
| store_db   |
| union_db   |
+-----+
```

1. Find the top 15 most common first names of the company's male's employees. Create a report showing the first name, its frequency, and its rank.

DATA DESCRIPTION:

Employee records are stored in the metastore table hr_records in the database hr_db.

OUTPUT REQUIREMENTS:

- Place the result data in the hdfs directory: /user/studentID/solution
- Use a text format with a hyphen as the columnar delimiter.
- Order results by rank in ascending order.
- No ranks should be skipped if there are ranks with multiple names.

```
scala> spark.sql("use hr_db")
res1: org.apache.spark.sql.DataFrame = []

scala> var nomor1 = spark.sql("""
  | select first_name, frequency, dense_rank() over (order by frequency desc) rank
  | from
  | (
  |   select first_name, count(*) frequency from hr_records
  |   where gender='M'
  |   group by first_name
  |   order by count(*) desc
  | )T
  | """)
nomor1: org.apache.spark.sql.DataFrame = [first_name: string, frequency: bigint ... 1 more field]

scala> nomor1.write.mode("overwrite").format("csv").option("delimiter", "-").option("header", "true").save("hdfs://user/2501967096/solution")

scala> spark.read.option("header", "true").csv("hdfs://user/2501967096/solution").show(15)
+-----+
|first_name-frequency-rank|
+-----+
|      Fidel-253-1|
|      Ralph-252-2|
|       Otis-250-3|
|Terrance-250-3|
|       Otha-249-4|
|       Lamar-247-5|
|Efrain-246-6|
|Alvaro-244-7|
|    Phil-243-8|
|Walker-243-8|
|    Keith-242-9|
|    Amos-242-9|
|    Myron-242-9|
|    Luigi-242-9|
|Garfield-241-10|
+-----+
only showing top 15 rows
```

```
verulam-blue ~ hdfs dfs -ls /user/2501967096/solution
Found 2 items
-rw-r--r-- 1 verulam-blue supergroup 0 2023-02-04 03:08 /user/2501967096/solution/_SUCCESS
-rw-r--r-- 1 verulam-blue supergroup 16738 2023-02-04 03:08 /user/2501967096/solution/part-00000-9a292037-9782-4603-b072-41b9a080678a-c000.csv
```

Note: Total records of final output should be 1219

```
scala> nomor1.count()
res4: Long = 1219
```

2. # INSTRUCTIONS:

Use the EMR data to find the total number of emergency department visits that were due to influenza-like illness and/or pneumonia that resulted in hospitalization for the months of May, June & July

DATA DESCRIPTION:

Emergency department visits records are stored as Parquet files, compressed using gzip, and stored in the HDFS directory. You may access the directory using this path:
/user/verulam_blue/data/emr_data

- The 'date_of_visit' column represents the date of hospital visit.
- The 'column li_pne_admissions' represent the count of influenza-like illness and/or pneumonia visits that went on to be admitted to the hospital.

OUTPUT REQUIREMENTS:

- Place the result data in the hdfs directory: /user/studentID/solution
- Results should show month of visit and number of hospitalizations.
- Use a text format with a tab as the columnar delimit

```
verulam-blue ~ hdfs dfs -ls /user/verulam_blue/data/emr_data
Found 1 items
-rw-r--r-- 1 verulam-blue supergroup 7588803 2020-10-13 06:39 /user/verulam_blue/data/emr_data/emr_flu_visits.gz.parquet

scala> var nomor2_read = spark.read.parquet("hdfs://user/verulam_blue/data/emr_data/emr_flu_visits.gz.parquet")
nomor2_read: org.apache.spark.sql.DataFrame = [extract_date: date, date_of_visit: date ... 4 more fields]

scala> nomor2_read.createOrReplaceTempView("table_no2")

scala> var nomor2 = spark.sql("""
  | select month(date_of_visit) as 'month_of_visit', sum(ili_pne_admissions) as 'number_of_hospitalizations' from table_no2
  | where month(date_of_visit)>=5 and month(date_of_visit)<=7
  | group by month(date_of_visit)
  | order by month(date_of_visit)
  | """)
nomor2: org.apache.spark.sql.DataFrame = [month_of_visit: int, number_of_hospitalizations: bigint]

scala> nomor2.write.mode("overwrite").format("csv").option("delimiter", "\t").option("header", "true").save("hdfs://user/2501967096/solution")

scala> spark.read.option("header", "true").csv("hdfs://user/2501967096/solution").show()
+-----+
|month_of_visit|number_of_hospitalizations|
+-----+
|5|200801|
|6|113289|
|7|122021|
+-----+

verulam-blue ~ hdfs dfs -ls /user/2501967096/solution
Found 4 items
-rw-r--r-- 1 verulam-blue supergroup 0 2023-02-04 03:11 /user/2501967096/solution/_SUCCESS
-rw-r--r-- 1 verulam-blue supergroup 51 2023-02-04 03:11 /user/2501967096/solution/part-00000-6ccb0e28-0078-4be3-a3c4-5ccc5d9dc30-c000.csv
-rw-r--r-- 1 verulam-blue supergroup 51 2023-02-04 03:11 /user/2501967096/solution/part-00001-6ccb0e28-0078-4be3-a3c4-5ccc5d9dc30-c000.csv
-rw-r--r-- 1 verulam-blue supergroup 51 2023-02-04 03:11 /user/2501967096/solution/part-00002-6ccb0e28-0078-4be3-a3c4-5ccc5d9dc30-c000.csv
```

3. Working with the "gp_db" database use the "items" column from the metastore table "gp_rx" together with the "gp_address" table to find the total number of prescriptions made by each GP practice in the city of Bolton.

- The "items" column, in the metastore table "gp_rx", represents the "total number of items prescribed".

- The first 4 letters of postcodes for GP practices in Bolton are: 'BL1 ', 'BL2 ', 'BL3 '

OUTPUT REQUIREMENTS

- Results data should be saved as a JSON file
- Place the result data in the hdfs directory: /user/studentID/solution
- Order results by "practice_code" in descending order.

```
scala> spark.sql("use gp_db")
res8: org.apache.spark.sql.DataFrame = []

scala> var nomor3 = spark.sql("""
| select gp_rx.practice_code, surgery_name, sum(items) nbr_prescriptions
| from gp_address
| join gp_rx
| on gp_address.practice_code=gp_rx.practice_code
| where substring(postcode,1,4) like "BL1 " or substring(postcode,1,4) like "BL2 " or substring(postcode,1,4) like "BL3 "
| group by gp_rx.practice_code, surgery_name
| order by gp_rx.practice_code desc
| """)
nomor3: org.apache.spark.sql.DataFrame = [practice_code: string, surgery_name: string ... 1 more field]
```

```
scala> nomor3.show()
+-----+-----+-----+
|practice_code|surgery_name|nbr_prescriptions|
+-----+-----+-----+
|Y04600|BARDOC GP OOH|3042|
|Y03641|BOLTON COMMUNITY ...|2267|
|Y03366|OLIVE FAMILY PRAC...|5030|
|Y03364|GREAT LEVER PRACTICE|5619|
|Y03079|BOLTON COMMUNITY ...|22209|
|Y02943|NEUROLOGY LONG TE...|56|
|Y02790|BOLTON MEDICAL CE...|4155|
|Y02319|BOLTON GENERAL PR...|5095|
|Y00747|HALLIWELL HEALTH ...|165|
|Y00552|MINOR TREATMENT C...|1|
|Y00448|DIABETES CENTRE|150|
|Y00233|THE PARALLEL|1|
|Y00215|ORTHOPAEDIC & RHE...|70|
|Y00208|TIER 2 DERMATOLOG...|9|
|Y00186|3D MEDICAL CENTRE|1547|
|P82661|INTERMEDIATE CARE|470|
|P82660|DEANE CLINIC 1|6620|
|P82654|BOLTON HOSPICE|14|
|P82640|AL FAL MEDICAL GROUP|6785|
|P82634|WYRESDALE ROAD SU...|5443|
+-----+-----+-----+
only showing top 20 rows
```

```
scala> nomor3.write.mode("overwrite").format("json").option("header", "true").save("hdfs:/user/2501967096/solution")
```

```
verulam-blue ~ hdfs dfs -ls /user/2501967096/solution
```

[illegible]