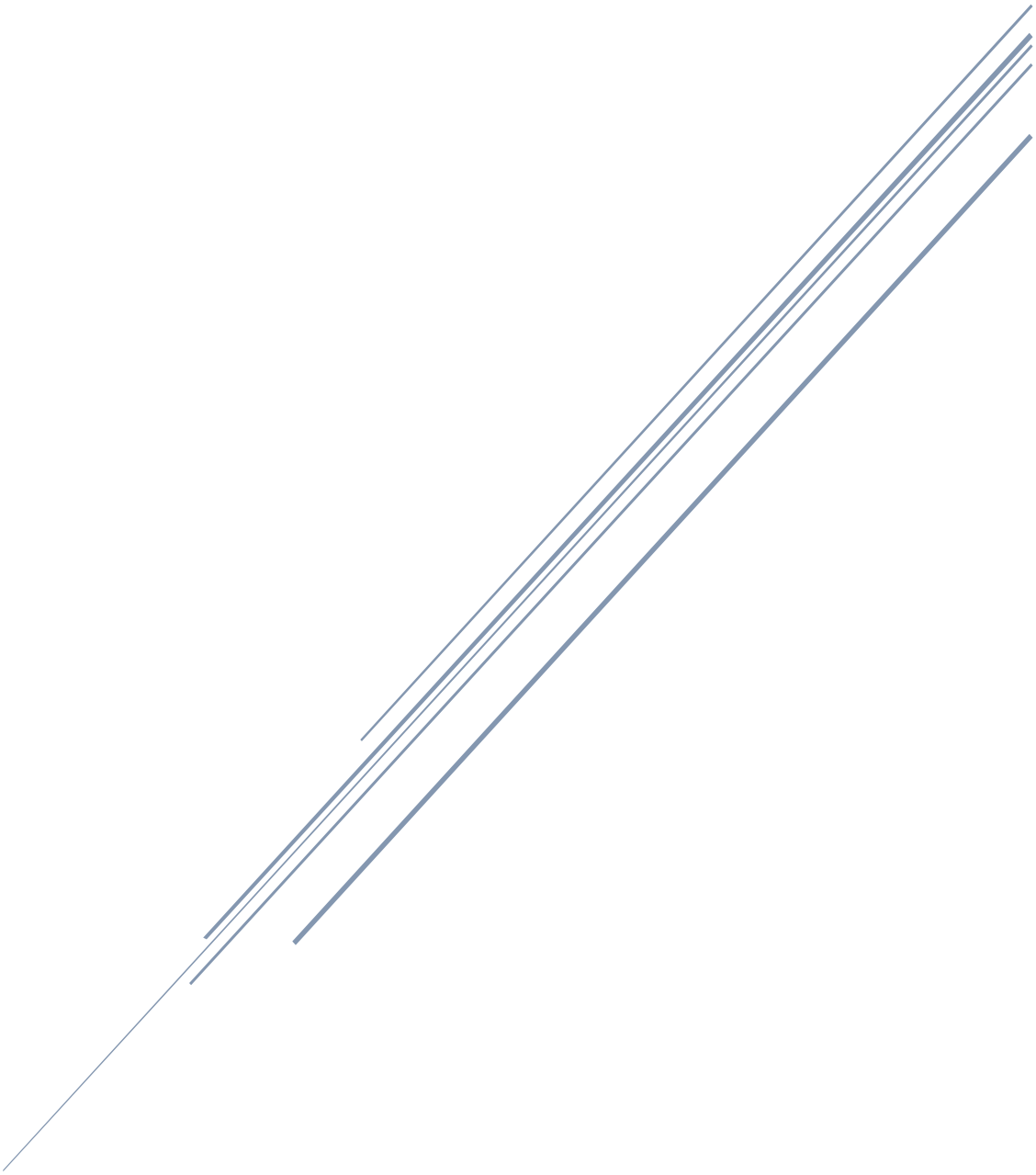
AWS ATHENA SCRIPTS

An addendum to Analysis of Medical Analysis



UNIVERSITY OF SALFORD
Big Data Tools & Techniques

Query 1 – Create database

```
-- DROP DATABASE coviddata CASCADE;


-- To create a CORD19 based database

CREATE DATABASE coviddb28;
```

Query 2 – Create CORD19 data table

```
-- To create a table for July dataset - CORD19 metadata

CREATE EXTERNAL TABLE IF NOT EXISTS coviddb28.c19_2020_07_01 (

  `cord_uid` string,

  `sha` string,

  `source_x` string,

  `title` string,

  `doi` string,

  `pmcid` string,

  `pubmed_id` string,

  `license` string,

  `abstract` string,

  `publish_time` string,

  `authors` string,

  `journal` string,

  `Microsoft_Academic_Paper_ID` string,

  `WHO_#Covidence` string,

  `arxiv_id` string,

  `has_pdf_parse` string,

  `has_pmc_xml_parse` string,

  `full_text_file` string,

  `url` string

)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES ("skip.header.line.count"="1", "separatorChar" = ",", "escapeChar" = "\"", "quoteChar" = "\"")

LOCATION 's3://jonnavittula.inputs-bucket/Prd1/'


-- Drop the table c19_2020_07_01 if necessary

-- DROP TABLE coviddb28.c19_2020_07_01
```

Query 3 – Create Scimago data table

```
CREATE EXTERNAL TABLE IF NOT EXISTS coviddb28.scimagojr19 (
Rank INT,
Sourceid STRING,
Title STRING,
Type STRING,
Issn STRING,
SJR STRING,
`SJR_Best_Quartile` STRING,
`H_index` INT,
`Total_Docs.(2019)` STRING,
`Total_Docs.(3years)` STRING,
`Total_Refs.` STRING,
`Total_Cites(3years)` STRING,
`Cites/Doc.(2years)` STRING,
`Ref./Doc.` STRING,
`Country` STRING,
Region STRING,
Publisher STRING,
Coverage STRING,
Categories STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES ("skip.header.line.count" = "1", "separatorChar" = '\073')
LOCATION 's3://jonnavittula.inputs-bucket/SJR/'
```

Query 4 – Create Views to clean Authors

```
DROP VIEW IF EXISTS Authors_list ;



-- View 1 - Authors split on ";"
CREATE OR REPLACE VIEW Authors_Raw AS
SELECT journal, SPLIT(REPLACE(authors, ',', '|'), ';') as authors_raw
FROM "coviddb28"."c19_2020_07_01" ;



-- View 2 - Authors Exploded View
CREATE OR REPLACE VIEW Authors_Cleaned AS
WITH dataset AS
 (
   SELECT journal, authors_raw FROM "coviddb28"."Authors_Raw"
 )
SELECT journal,
      REPLACE(TRIM(Author_Piped), '|', ',') as Author
FROM dataset
CROSS JOIN UNNEST(authors_raw) as t(Author_Piped)
;
```

Query 5 – Task 1 – View (refer in AWS Athena)

```
-- Task 1. Find the 5 most common journals, list them along with their frequencies.

CREATE OR REPLACE VIEW cord19_task1 AS
SELECT
  journal AS Journal,
  count(journal) AS Frequency
FROM
  "coviddb28"."c19_2020_07_01"
WHERE
  length(journal) > 0
GROUP BY
  journal
ORDER BY
  Frequency DESC
LIMIT
  5
```

Query 6 – Task 2 – View (refer in AWS Athena)

```
-- Task 2. The top 5 average abstract lengths (number of words) per journal.


CREATE OR REPLACE VIEW cord19_task2 AS
SELECT
 journal AS Journal,
 ROUND(
  AVG(
   LENGTH(abstract) - LENGTH(REPLACE(abstract, ' ', '')) + 1
    ),1) AS Average_Abstract_Lengths
FROM
 "coviddb28"."c19_2020_07_01"
WHERE
  abstract <> ‘’
GROUP BY
 journal
ORDER BY
 Average_Abstract_Lengths DESC
LIMIT
 5
```

Query 7 - Task 3 – View (refer in AWS Athena)

```
-- Task 3. Titles of the 5 papers with the highest numbers of authors. Both the numbers of authors and the

-- corresponding titles need to be output.


CREATE OR REPLACE VIEW cord19_task3 AS

SELECT

  title as Title,

  SUM(length(authors) - LENGTH(REPLACE(authors, ';', '')) +1 ) AS Num_of_Authors

FROM

     "coviddb28"."c19_2020_07_01"

GROUP BY

  title

ORDER BY

  Num_of_Authors DESC

LIMIT

  5
```

Query 8 – Task 4 – View (refer in AWS Athena)

```
-- Task 4. The top 5 most prolific authors along with the number of papers they have contributed to.

CREATE OR REPLACE VIEW cord19_task4 AS
SELECT author as Author,
    COUNT(journal) as Num_of_Papers
FROM "coviddb28"."authors_cleaned"
WHERE author <> '' AND SUBSTR(author,2,3) <> '039'
GROUP BY author
ORDER BY Num_of_Papers DESC
LIMIT 5;
```

Query 9 – Task 5 – View (refer in AWS Athena)

-- Task 5. If an author's H index is computed by summing all the H indexes of the journals they've published

-- in (as included in the scimagojr dataset), list the 5 people with the top author H index values.

```
CREATE OR REPLACE VIEW cord19_task5 AS
SELECT author,
    SUM(H_index) as Total_H_Index
FROM coviddb28.authors_cleaned as C
JOIN coviddb28.scimagojr19 as S
ON C.journal = S.Title
WHERE length(author) > 0
GROUP BY author
ORDER BY Total_H_Index DESC
LIMIT 5
```

Query 10 – Task 6 – View (refer in AWS Athena)

```
-- Task 6. Plot the number of papers per month since 2020-01. You need to include your visualization as well

-- as a table of the values you have plotted for each month.


CREATE OR REPLACE VIEW cord19_task6 AS

SELECT

  SUBSTR(publish_time, 1, 7) as Year_Month,

  COUNT(*) as Published_Count

FROM

  "coviddb28"."c19_2020_07_01"

WHERE

  publish_time LIKE '2020%' AND publish_time <> '2020'

GROUP BY

  SUBSTR(publish_time, 1, 7)

ORDER BY

  SUBSTR(publish_time, 1, 7) ASC
```