**Problems when building Cassandra database based on querys**

**In the Cassandra Modeling section, I split the whole data into different parts which are based on our queries.**

Query1: Select all records that has a story title that contains "politics" by from 2015 to 2018. Show the source, author, story title and year of publication

Read: SELECT \* FROM politics\_by\_year WHERE year>2014 and year<2019 ALLOW FILTERING;

Problem: Can not order in the right order

Outcome:A screenshot of a cell phone

Description automatically generated

SELECT \* FROM politics\_by\_year WHERE year>2014 and year<2019

ORDER BY year

ALLOW FILTERING;

Query2: Select all records that has a story title that contains "facebook", "intel", Microsoft". Show the average, maximum, minimum scores of each.

Read: SELECT title, year, min/max/avg(score) AS score, authors\_username, story\_id FROM name\_score WHERE name=1/2/3;

Problem: Can not select the title by using command such as “like ‘%facebook%’” in sql.

Solution: add another column called ‘name’ and classified all the titles contain “Facebook” into number 1, “Intel” into number 2, “Microsoft” into number 3.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A picture containing indoor

Description automatically generated

Outcome:

Facebook

Max

A screenshot of a cell phone

Description automatically generated

MinA screenshot of a cell phone

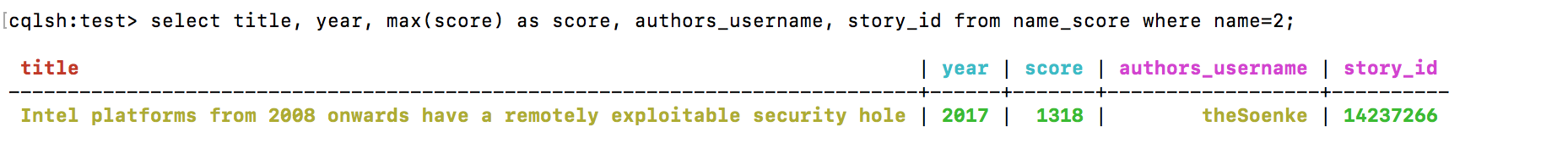
Description automatically generated

Avg

A screenshot of a cell phone

Description automatically generated

Intel

Max

MinA screenshot of a cell phone

Description automatically generated

Avg

A screenshot of a cell phone

Description automatically generated

Microsoft maxA screenshot of a cell phone

Description automatically generated

Min

A screenshot of a cell phone

Description automatically generated

avgA screenshot of a cell phone

Description automatically generated

Query3: Select the lowest scores of each year from 2013 to 2018 . Show the story title and author.

Read: SELECT MIN (Score) AS Score, Title, Author\_Username, Year, Story\_ID FROM lowest\_score\_by\_year WHERE Year=2013/2014/2015/2016/2017/2018 ALLOW FILTERING;

Problem: Data of 2009 and 2012 can not be loaded into Cassandra, so I changed the query into ‘Select the lowest scores of each year from 2013 to 2018’A close up of a piece of paper

Description automatically generated

Outcome:

A screenshot of a cell phone

Description automatically generated

Query4: Show all authors who created stories more than 30 times . Show the author.

Read: SELECT Authors\_username, Count FROM authors\_30 WHERE Count>30 ALLOW FILTERING;

Problem: Can not use Count() command to summarize the appearing time of each author. So I replaced the method with using Count() command in SQL then loading the data into Cassandra.

A screenshot of a cell phone

Description automatically generated

Outcome:

A screenshot of a social media post

Description automatically generated

Query5: Select the latest story created, show the date of publication, story\_id, title, and author

Read: SELECT Title, Story\_ID, Authors\_username, Year, Quarter, Month From latest\_story WHERE Year>2017 and Quarter>3 and Month>11 ALLOW FILTERING;

Outcome:A screenshot of a cell phone

Description automatically generated

1.COPY the csv files from local into the docker container: docker cp [csv file path] [container name]:rootA close up of a logo

Description automatically generated

2.Then execute your container : docker exec -it [container name] bash

3.Get into root container: cd root

4.check if the csv file is copied successfully:lsA screenshot of a cell phone

Description automatically generated

5.use ‘cqlsh’ to run CQL shell and run command ‘use keyspace’s name’ to use the keyspace.

A screenshot of a cell phone

Description automatically generated

6.finally, you can copy the data from the csv file. With command: COPY [table name] FROM ‘[csv file name.csv]’ WITH HEADER=TRUE;