**Homework Assignment 5**

Due: 11:59PM June 22, 2021

1. Navigate the data

(a) Execute the following queries (on either Hive or Impala) and report the results.

- SHOW DATABASE;

테이블이(가) 표시된 사진

자동 생성된 설명

- SELECT \* FROM toy.toys;

테이블이(가) 표시된 사진

자동 생성된 설명

- SELECT CONCAT(first\_name, ' ', last\_name) AS NAME, office\_id

FROM default.employees;

테이블이(가) 표시된 사진

자동 생성된 설명

- SELECT \* FROM default.orders WHERE total > 25;

테이블이(가) 표시된 사진

자동 생성된 설명

- SELECT employees.last\_name, employees.office\_id, city, state\_province, country

FROM employees JOIN offices ON employees.office\_id = offices.office\_id;

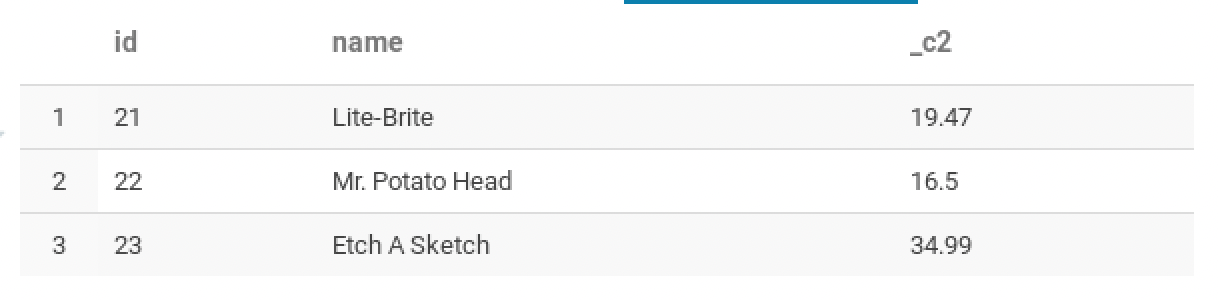
테이블이(가) 표시된 사진

자동 생성된 설명

(b) Execute the following queries on both Hive and Impala, and explain the difference in terms of the results and performances.

- SELECT id, name, price + 5 FROM toy.toys;

(Hive)



(Impala)

테이블이(가) 표시된 사진

자동 생성된 설명

(difference in terms of the results and performances)

Hive doesn't show the same data, Impala shows everything. Also, the Impala has much faster performance.

- SELECT id, name, ROUND(price\*1.1,2) FROM toy.toys;

(Hive)

텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명

(Impala)

텍스트이(가) 표시된 사진

자동 생성된 설명

(difference in terms of the results and performances)

Impala is faster than Hive. And the column name can change in Impala.

- SELECT COUNT(\*) FROM wax.crayons;

(Hive)

테이블이(가) 표시된 사진

자동 생성된 설명

(Impala)



(difference in terms of the results and performances)

Impala is faster than Hive. And have different column name.

- SELECT red + green + blue AS SUM\_RGB

FROM wax.crayons ORDER BY SUM\_RGB DESC LIMIT 10;

(Hive)



(Impala)



(difference in terms of the results and performances)

Impala is faster than Hive.

- SELECT order\_id, name AS cust\_name, last\_name AS empl\_name

FROM default.orders

JOIN default.customers ON orders.cust\_id = customers.cust\_id

JOIN default.employees ON orders.empl\_id = employees.empl\_id

WHERE total < 0;

(Hive)

텍스트이(가) 표시된 사진

자동 생성된 설명

(Impala)

텍스트이(가) 표시된 사진

자동 생성된 설명

(difference in terms of the results and performances)

Impala is faster than Hive.

- SELECT order\_id, name AS cust\_name, last\_name AS empl\_name

FROM default.orders

JOIN default.customers ON orders.cust\_id = customers.cust\_id

JOIN default.employees ON orders.empl\_id = employees.empl\_id

ORDER BY total;

(Hive)

텍스트이(가) 표시된 사진

자동 생성된 설명

(Impala)

테이블이(가) 표시된 사진

자동 생성된 설명

(difference in terms of the results and performances)

Hive doesn’t allow order by. However, Impala allow order by.

- difference between Hive and Impala

From a result & performance point of view, there is a noticeable difference between impala and hive.

First, impala is a system that can analyze data stored in HDFS in real time using SQL. Therefore, it provides very fast results because analysis is performed using a distributed query engine without using the MapReduce framework.

In addition, the CPU load is reduced and the I/O response time is reduced, so the performance is also very good.

However, Hive uses the Mapreduce framework and is about data inquiry interface, so the speed of data analysis work is not high. Therefore, it gives slower results compared to impala.

2. From 3(a) of the worksheet, report the result of each query on Impala. Add your analysis & interpretation about the results.

(a) GROUP BY queries

- SELECT year, COUNT(\*), ROUND(AVG(dep\_delay),1), ROUND(AVG(arr\_delay),1)

FROM flights

GROUP BY year

ORDER BY year;

테이블이(가) 표시된 사진

자동 생성된 설명

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query means average dep\_delay and average arr\_delay during flight during the same year. Based on this, it can be seen that the overall delay in 2012 is small and dep\_delay is larger than arr\_delay.

- SELECT month, COUNT(\*), ROUND(AVG(dep\_delay),1), ROUND(AVG(arr\_delay),1)

FROM flights

GROUP BY month

ORDER BY month;

테이블이(가) 표시된 사진

자동 생성된 설명

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query returns the number of flights, average dep\_delay, and average arr\_delay for the same month. Based on this, it can be seen that November has the smallest delay, and similarly to the previous problem, dep\_delay is larger than arr\_delay.

- SELECT FLOOR(dep\_time/100) AS DEP\_HR, COUNT(\*), ROUND(AVG(dep\_delay),1)

FROM flights

GROUP BY DEP\_HR

ORDER BY DEP\_HR;

테이블이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query shows the number of delays and dep\_delay time in the same time zone.

Based on this, the average number of delays the most is in the 8am time zone, but the longest delay time on average is 2am.

- SELECT origin, COUNT(\*), ROUND(AVG(dep\_delay),1) AS DEP\_DLY

FROM flights

GROUP BY origin

ORDER BY DEP\_DLY DESC;

테이블이(가) 표시된 사진

자동 생성된 설명ans so on…

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query shows the number of flights to each region and the average dep\_delay time. Based on this, ENW, SWO, TKI, and BFF have the smallest number of delays with 1, and ACV has the largest number of delays. On the contrary, it can be seen that the number of delays and the average delay time are inversely proportional. It may mean that the number of planes is small, and thus the number of customers who use the planes may also be small.

- SELECT carrier, ROUND(AVG(dep\_delay),1) AS DEP\_DELAY,

ROUND(AVG(arr\_delay),1) AS ARR\_DELAY

FROM flights

GROUP BY carrier

ORDER BY ARR\_DELAY DESC;

테이블이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query shows the average dep\_delay time and average arr\_delay time of each carrier in descending order of arr\_delay. When analyzing the results based on this, if dep\_delay is longer than arr\_delay, it means that the plane was definitely overspeeding or it took a little less than the original flight time. It can also be seen that overall dep\_delay is longer than arr\_delay.

- SELECT carrier, ROUND(AVG(dep\_delay),1) AS DEP\_DELAY,

ROUND(AVG(arr\_delay),1) AS ARR\_DELAY

FROM flights

GROUP BY carrier

ORDER BY DEP\_DELAY DESC;

테이블이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

(analysis & interpretation about results)

impala provides round() function. However, as a result of searching on Google, it seems that it is not displayed properly due to a client problem. This query shows the average dep\_delay time and average arr\_delay time of each carrier in descending order of dep\_delay. When analyzing the results based on this, if dep\_delay is longer than arr\_delay, it means that the plane was definitely overspeeding or it took a little less than the original flight time. It can also be seen that overall dep\_delay is longer than arr\_delay.