This short project is done in Cloudera QuickStart VM using Apache Hive

Question:

Suppose you have a dataset like the following:

Process_date	id1	id2	count
2017-05-01	T	S	1
2017-05-03	T	S	1
2017-05-04	T	S	1
2017-06-01	T	S	2
2017-07-01	T	S	1
2017-07-02	T	Z	1
2017-07-03	T	Z	1
2017-08-04	B	Z	1

Given the data above, please write a SQL query which groups the data by columns ***id1*** and ***id2***. Your query should also take into account the value in the ***count*** column so that when it changes, a new group is formed. For example, In the 4th row, the ***count*** value changes from 1 to 2, therefore, a new group is expected to form here for the date 2017-06-01.

For each group we would like to have max_process_date, min_process_date, id1, id2 and count columns.

To clarify through the same example as above, the result of running your query on the data above should produce the following table:

Id1	id2	count	min_process_date	max_process_date
T	S	1	2017-05-01	2017-05-04
T	S	2	2017-06-01	2017-06-01
T	S	1	2017-07-01	2017-07-01
T	Z	1	2017-07-02	2017-07-03
B	Z	1	2017-08-04	2017-08-04

Answer:

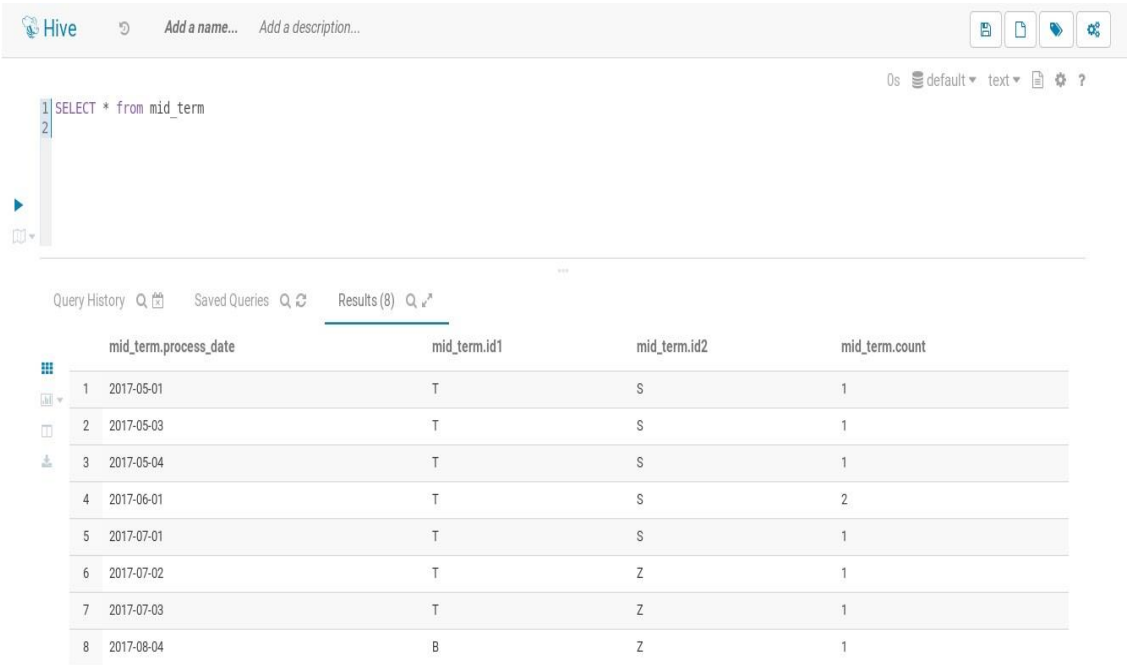
Step 1:

Main table (named as mid_term) was added to the database. And then select everything to show what values do we have in our table.

SQL Query:

```
SELECT * from mid_term
```

Output:



	mid_term.process_date	mid_term.id1	mid_term.id2	mid_term.count
1	2017-05-01	T	S	1
2	2017-05-03	T	S	1
3	2017-05-04	T	S	1
4	2017-06-01	T	S	2
5	2017-07-01	T	S	1
6	2017-07-02	T	Z	1
7	2017-07-03	T	Z	1
8	2017-08-04	B	Z	1

Step 2 (Final):

Required Output

SQL Query:

```
SELECT id1,id2,count,MIN(process_date) AS min_process, Max(process_date) AS max_process
FROM
 (
  select id1,id2,process_date, count,
    -- assign a number to all rows with the same value
    sum(flag) over (order by process_date rows unbounded preceding) as grp
  from
   ( SELECT id1,id2,process_date, count,
       -- assigns 1 whenever count changes
       case when lag(count) over (order by process_date) = count
           then 0 -- value of the previous same as current
           else 1 -- different value
       end as flag
     from mid_term
   ) as process_date
 ) as process_date
GROUP BY id1,id2,count, grp
ORDER BY min_process,max_process
```

Output:

Hive Add a name... Add a description...

1m, 42s default text

```
SELECT id1,id2,count,MIN(process_date) AS min_process, Max(process_date) AS max_process
FROM
 (
   select id1,id2,process_date, count,
     -- assign a number to all rows with the same value
     sum(flag) over (order by process_date rows unbounded preceding) as grp
   from
     ( SELECT id1,id2,process_date, count,
         -- assigns 1 whenever count changes
         case when lag(count) over (order by process_date) = count
              then 0 -- value of the previous same as current
              else 1 -- different value
         end as flag
       from mid_term
     ) as process_date
  ) as process_date
GROUP BY id1,id2,count, grp
ORDER BY min_process,max_process
```

Query History Saved Queries Results (5)

	id1	id2	count	min_process	max_process
1	T	S	1	2017-05-01	2017-05-04
2	T	S	2	2017-06-01	2017-06-01
3	T	S	1	2017-07-01	2017-07-01
4	T	Z	1	2017-07-02	2017-07-03
5	B	Z	1	2017-08-04	2017-08-04