

SQL Developer Questionnaire

1. Given a table with a `original_date` column of string type, write **SQL**(hive preferably) code to calculate two more columns –

`end_of_week` - to calculate the end of week date i.e. date of next Sunday from that date, unless that date is a Sunday, then it is the same date

`end_of_month` - to calculate the end of month date

Example:

ORIGINAL_DATE	END_OF_WEEK	END_OF_MONTH
20190828	20190828	20190831
20190829	20190904	20190831
20190830	20190904	20190831
20190831	20190904	20190831
20190901	20190904	20190930
20190902	20190904	20190930
20190903	20190904	20190930

Explanation:

20190828 is a Sunday, so the `end_of_week` for the week is still that same date.

20190830 is a Tuesday, so the next Sunday is on 20190904

SOLUTION -

- Given a table `Activity`, with a `start_time` and `end_time`. Write **SQL**(hive preferably) code to assign a `group_id` for every overlapping intervals

Example:

ID	START_TIME	END_TIME	GROUP_ID
100	10:00	12:00	1
100	10:15	12:30	1
100	12:15	12:45	1
100	13:00	14:00	2
200	10:15	10:30	3

Explanation: for a given `id`, if the intervals overlap in any way i.e. if the `start_time` or `end_time` is in between any of the other record's `start_time` and `end_time` and belonging to the same `id`, then they should be part of the same group, hence should have the same `group_id`.

In our example, row1 has an interval from 10:00 to 12:00 which has overlapping interval with row2 because 10:15 is in between 10:00 and 12:00. Row3 has a start time of 12:15 which is overlapping with row2's intervals of 10:15 to 12:30. Therefore, row1, row2 and row3 belong to the same group. However, row4 doesn't share its intervals with any of row1, row2 or row3's intervals and so belongs to a different group. Row5 has a different id from the rest of the rows, so it is part of its own group.

Acceptable solution: As long as the `group_id` is unique for every different `id` and it is same for any overlapping group of intervals

SOLUTION -