SQL Scripts of view ‘v_att_data’ (removing the duplicated transitions with the little difference of time)

```
CREATE OR REPLACE VIEW V_ATT_ALL AS
with categorized_transitions as (
SELECT v.id,
       v.emp_id,
       v.emp_name,
       TRUNC(v.event_date) AS work_date,
       v.event,
       v.time_part,
       cp.name check_point,
       cp.start_time,
       cp.end_time,
       ROW_NUMBER() OVER(PARTITION BY v.emp_id, v.date_part, v.event ORDER BY v.time_part) AS
row_num
   FROM v_att_data v
   join checkpoints cp
     on to_timestamp(v.time_part, 'HH24:MI:SS') between
       to_timestamp(cp.start_time, 'HH24:MI:SS') and
       to_timestamp(cp.end_time, 'HH24:MI:SS')
  where v.event IN ('IN', 'OUT')),

  next_event as(

  select
  c1.emp_id,
  c1.emp_name,
  c1.work_date,
  c1.event,
  c1.time_part,
```

```
  c1.check_point,
  c1.start_time,
  c1.end_time,
  LEAD(c1.event) OVER (PARTITION BY c1.emp_id, c1.work_date ORDER BY c1.time_part) AS next_event,
  LEAD(c1.time_part) OVER (PARTITION BY c1.emp_id, c1.work_date ORDER BY c1.time_part) AS
next_time,
  LEAD(c1.check_point) OVER (PARTITION BY c1.emp_id, c1.work_date ORDER BY c1.time_part) AS
next_checkpoint
  from categorized_transitions c1),

 in_out_list as(
  SELECT c.emp_id,
     c.work_date,
     MAX (CASE
          WHEN c.event = 'OUT' and c.check_point = 'morning' then
              case when c.next_event = 'IN' then c.next_time
                  else null end
          END
       ) AS first_in,

     MIN(CASE
          WHEN event = 'IN' THEN
           c.time_part
        END) AS first_in2,

     MAX(CASE
          WHEN event = 'OUT' AND c.check_point = 'lunch' AND
              to_timestamp(c.time_part, 'HH24:MI:SS') BETWEEN
              to_timestamp(c.start_time, 'HH24:MI:SS') AND
```

```
            to_timestamp(c.end_time, 'HH24:MI:SS') THEN time_part
          WHEN event = 'OUT' AND c.check_point = 'working_am' AND c.next_event = 'IN' AND
c.next_checkpoint = 'working_am'
                    and to_timestamp(c.time_part, 'HH24:MI:SS') BETWEEN
                    to_timestamp(c.end_time, 'HH24:MI:SS')- interval '2' hour AND
                    to_timestamp(c.end_time, 'HH24:MI:SS') then null
          WHEN event = 'OUT' and c.check_point = 'working_am' AND c.next_checkpoint <> 'working_am'
                    and to_timestamp(c.time_part, 'HH24:MI:SS') BETWEEN
                    to_timestamp(c.end_time, 'HH24:MI:SS')- interval '2' hour AND
                    to_timestamp(c.end_time, 'HH24:MI:SS') then time_part

        END) AS lunch_out,

     MIN(CASE
        WHEN event = 'IN' AND c.check_point = 'lunch' AND
           to_timestamp(c.time_part, 'HH24:MI:SS') BETWEEN
           to_timestamp(c.start_time, 'HH24:MI:SS') AND
           to_timestamp(c.end_time, 'HH24:MI:SS')+ interval '30' minute THEN time_part
        WHEN event = 'IN' AND c.check_point = 'working_pm' AND
           to_timestamp(c.time_part, 'HH24:MI:SS') BETWEEN
           to_timestamp(c.start_time, 'HH24:MI:SS') AND
           to_timestamp(c.start_time, 'HH24:MI:SS')+ interval '2' hour THEN time_part
        END) AS lunch_in,
     MAX(CASE
        WHEN event = 'OUT' THEN
         c.time_part
        END) AS last_out
  FROM next_event c
 GROUP BY c.emp_id, c.work_date)
```

```
select
cc.id event_id,
cc.emp_id,
cc.emp_name,
cc.work_date,
cc.event,
cc.time_part,
cc.check_point,
cc.start_time,
cc.end_time,
vio.detection ,
case
  when cc.event = 'IN' and to_timestamp(cc.time_part , 'HH24:MI:SS') = to_timestamp(l.first_in ,
'HH24:MI:SS') then cc.time_part
    else null end first_in,
case
  when cc.event = 'IN' and to_timestamp(cc.time_part , 'HH24:MI:SS') = to_timestamp(l.first_in2 ,
'HH24:MI:SS') then cc.time_part
    else null end first_in2,
case
  when cc.event = 'OUT' and to_timestamp(cc.time_part , 'HH24:MI:SS') = to_timestamp(l.lunch_out ,
'HH24:MI:SS') then cc.time_part
    else null end lunch_out,
case
  when cc.event = 'IN' and to_timestamp(cc.time_part , 'HH24:MI:SS') = to_timestamp(l.lunch_in ,
'HH24:MI:SS') then cc.time_part
    else null end lunch_in,
case
```

```
  when cc.event = 'OUT' and to_timestamp(cc.time_part , 'HH24:MI:SS') = to_timestamp(l.last_out , 'HH24:MI:SS') then cc.time_part
    else null end last_out,
row_number() over (partition by cc.work_date,cc.emp_id order by cc.work_date) rn
from categorized_transitions cc
join in_out_list l on cc.emp_id = l.emp_id and cc.work_date = l.work_date
join violations vio on cc.id = vio.event_id
where 1 = 1
order by cc.emp_id,cc.work_date,cc.time_part;
```