Charalampos Chrysanthakopoulos – *updated solution*

We wish to calculate the monthly average booking revenue using the following formula :

$$monthly\_average\_booking\_revenue = \frac{total\ monthly\ revenue}{number\ of\ days\ in\ the\ month}$$

SQL code (*written for Oracle*):

```sql
-- CTE (d) gets month and year values from table property_days
with d as (
 select distinct
  building_code,
  to_char(date,'mm') as m,
  to_char(date,'yyyy') as y
 from property_days
 where
  is_active = 'TRUE'                       -- consider only active properties
  and date >= '01/01/2022'                 -- calendar start date
  and date < trunc(sysdate,'mm')           -- exclude running month from calculations
),

-- CTE (ds) calculates the number of days per month based on subquery (d)
ds as (
 select
  building_code, m, y,
  last_day(to_date(m||'/'||y, 'mm/yyyy')) - to_date(m||'/'||y, 'mm/yyyy') + 1 as nbr_days
 from d),

-- CTE (rev) calculates the total monthly revenue
rev as (
 select
  building_code,
  to_char(date,'mm') m,
  to_char(date,'yyyy') y,
  sum(revenue) as revenue
 from property_days
 where
  is_active = 'TRUE'                       -- consider only active properties
  and date >= '01/01/2022'                 -- calendar start date
  and date < trunc(sysdate,'mm')           -- exclude running month from calculations
 group by building_code, to_char(date,'mm'), to_char(date,'yyyy')
)

-- final query calculates the monthly average revenue
 select
  ds.y||'-'||ds.m||'-01' as month,
  ds.building_code,
  rev.revenue/ds.nbr_days as monthly_avg_revenue
 from ds, rev
 where
  ds.m = rev.m                               -- 1st join between (ds) and (rev)
  and ds.y = rev.y                           -- 2nd join between (ds) and (rev)
  and ds.building_code = rev.building_code   -- 3rd join between (ds) and (rev)
```