

SQL test 2



רקע

אתם עובדים באחת מחברות ההיטק המובילות בתחום הפינטק, לאחרונה פנה אליכם מנהל המוצר איתו אתם עובדים והוא בקש שתתנו לו גישה לדאטה, כך שהוא יוכל להציג אותה להנהלה. מכיוון שכמות הדאטה היא ענקית, חשבתם שהפתרון הכי טוב הוא לעשות מניפולציות על הדאטה לפני שאתם שולחים לו כי אין סיכוי שהוא יסתדר עם מאות אלפי שורות של דאטה באקסל שלו

זמן סביר לפתור מבחן זה הוא באזור ה-30 דקות עד שעה,

רשמו לעצמכם כמה זמן לקח לכם?

שאלות

לפניכם טבלה בשם, לכל שאלה יותר מפתרון אחד

revenue_data

country	revenue	date
Canda	120	2/1/21
Usa	145	3/4/21
Italy	234	3/4/21

1. עליכם ליצור את הטבלה הבאה כך שכל עמודה מייצגת את החודש הרלוונטי

country	January	February	March	April
Canda	120	120	240	240
Usa	145	290	145	145
Italy	100	70	35	35

2. חשבו לכל שבוע את הטרנד של הממוצע של השלושה שבועות אחרונים

revenue_moving_average	revenue	week
	100	1
	200	2
200	300	3
300	400	4
400	500	5
500	600	6

3. מסתבר שחסרים תאריכים של שבוע מסויים ובעקבות זאת הגרף נראה שבור, כיצד תוכל לתקן את הגרף באמצעות שאילתת sql

תשובות

שאלתא הכנה

```
select distinct extract(month from date) month,
               extract(year from date ) as year
  from `cs.country_data`
 order by 2,1

-- option 1
select
    country,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='09-20' then
amount end) as date_09_20,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='10-20' then
amount end) as date_10_20 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='11-20' then
amount end) as date_11_20 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='12-20' then
amount end) as date_12_20 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='01-21' then
amount end) as date_01_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='02-21' then
amount end) as date_02_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='03-21' then
amount end) as date_03_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='04-21' then
amount end) as date_04_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='05-21' then
amount end) as date_05_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='06-21' then
amount end) as date_06_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='07-21' then
amount end) as date_07_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='08-21' then
amount end) as date_08_21 ,
    sum(case when FORMAT_DATE("%m-%g",order_date) ='09-21' then
```

```

amount end) as date_09_21 ,
        sum(case when FORMAT_DATE("%m-%g",order_date) = '10-21' then
amount end) as date_10_21 ,
        sum(case when FORMAT_DATE("%m-%g",order_date) = '11-21' then
amount end) as date_11_21 ,
        sum(case when FORMAT_DATE("%m-%g",order_date) = '12-21' then
amount end) as date_12_21 ,
        from `course-349919.dwh.table_test_2`
        group by 1
        order by 2,1

```

```

-- option 2
DECLARE list_of_dates string;
SET list_of_dates = (select
        CONCAT('(', STRING_AGG(DISTINCT FORMAT_DATE("date_%m_%g",date),
        ',', ''), ')'),
        from `cs.country_data`
);

CREATE OR REPLACE TABLE `daidsql.cs.country_data_new`
as
(select * ,FORMAT_DATE("date_%m_%g",date) as new_date from
`cs.country_data`
);

EXECUTE IMMEDIATE format("""select * from
        (select country,new_date,amount from
`daidsql.cs.country_data_new`) \n pivot(sum(amount) FOR new_date in %s)
        """,list_of_dates);

```

שאלה 2

```

with t1 as
(

```

```
select * ,row_number() over(order by year,week ) as rn
  from
    (
select
  EXTRACT(year from date) year,
  EXTRACT(week from date) as week,
  sum(amount) as revenue
  from `cs.country_data`
  group by 1,2
  order by 1,2
)
)

select
  a.year,
  a.week,
  sum(a.revenue) as revenue,
  sum(b.revenue + c.revenue +d.revenue)/3 last_3_weeks_avg
  from t1 as a
  left join t1 as b
  on a.rn-1 = b.rn
  left join t1 as c
  on a.rn-2 = c.rn
  left join t1 as d
  on a.rn-3 = d.rn
  group by 1,2
  order by 1,2
```

Option 2

```
with t1 as
(
select * ,row_number() over(order by year,week ) as rn
  from
    (
select
  EXTRACT(year from date) year,
  EXTRACT(week from date) as week,
  sum(amount) as revenue
  from `cs.country_data`
```

```
group by 1,2
order by 1,2
)
)

select year,
       week,
       revenue,
       (previous+second_previous+third_previous)/3 avg_last_3_weeks
from
  (
select
  a.year,
  a.week,
  rn,
  a.revenue as revenue,
  lag(revenue,1) over (order by rn) previous,
  lag(revenue,2) over (order by rn) second_previous,
  lag(revenue,3) over (order by rn) third_previous,
from t1 as a
)
order by 1,2
```

