Continuous 13-weeks inactivity Calculation

Retail Year, Retail week , Seller , GMV, sellers migrated week (min → officially inactive)
2025,25,142525,1000, 3
2025,26,142525,1200,3

Check in Next 26 weeks whether seller has atleast 13 weeks of continuous inactivity until end of period

—---

More context on this →

Because businesses want to confirm *long-term inactivity*, not temporary pauses.

- 13 weeks = 1 quarter (minimum inactivity definition)
- 26 weeks = 2 quarters (confirmation window)

A seller must remain inactive long enough to qualify as fully inactive.

—---

Note → It is possible that the seller has not listed for a continuous consecutive 13 weeks

Interesting Insights: →

1.) Inactive_13_weeks = 13 by itself does not guarantee that the **13 weeks are calendar continuous**
2.) It only guarantees 13 rows in a window

week	gmv
10	0
11	0
14	0
15	0
16	0

From interview: Week 3 & Week 29th Migration*
Breakdown: →

1. Weekly data
2. Inactive data
3. Continuous 13-week data (Rolling sum)

4. Migrated 13 weeks (?) → first/min week where seller becomes officially inactive; property of seller
    a. Date - date*r_no’1DAY’INTERVAL
    b. 1 12 1 11
    c. 1 13 2 11
    d. 1 14 3 11
    e. Group by seller_id, grp → seller_id, min(dt)/min(retail_year-retail_week) as migrated_week//official inactive week
    f. Or sum() over(unbounded preceding ..) → inactivity_counter = 1 is 1st migration(retail_year & retail_week)
5. Check in the Next 26 weeks whether the seller has at least 13 weeks of continuous inactivity until the end of the period

—---

Soln: →

```
-- Weekly data
WITH revised_trans_data AS (
   SELECT
      dt,
      DATE_PART('year', dt) AS retail_year,
      DATE_PART('week', dt) AS retail_week,
      seller_id,
      SUM(gmv) AS gmv
   FROM trans_data
   GROUP BY 1,2,3,4
),

-- Step 1: Flag inactive & compute lag
flagged AS (
   SELECT
      seller_id,
      retail_year,
      retail_week,
      dt,
      gmv,
      CASE WHEN gmv = 0 THEN 1 ELSE 0 END AS inactive,
      row_number() over(partition by seller_id order by retail_year, retail_week) as seq_week
   FROM revised_trans_data
),
-- Step 2: Compute rolling 13-week inactivity
rolling13 AS (
   SELECT
      seller_id,
      retail_year,
        retail_week,
        dt,
        gmv,
        inactive,
        SUM(inactive) OVER (
            PARTITION BY seller_id
            ORDER BY retail_year, retail_week
            ROWS BETWEEN 12 PRECEDING AND CURRENT ROW
        ) AS inactive_13_weeks
    FROM flagged
),
-- Step 3: Determine migration flag + first migration week
first_migration AS (
    SELECT
        seller_id,
        MIN(retail_week) AS migrated_week
    FROM streaks
    WHERE inactive_13_weeks = 13
    GROUP BY seller_id
)
SELECT *
FROM first_migration;
```

Note → This is still not continuous. It's just consecutive ones after the other. For continuous explicit check of curr_dt - prev_dt = 1

2nd interesting part to the Question →

Check in Next 26 weeks whether seller has atleast 13 weeks of continuous inactivity until end of period

Breakdown →

1. 2 more quarters (Next 26 weeks)
    a. Inactivity started(seq_week)
    b. Next_26_table
    c. Between start_date(inactivity_seq_week1) and next_26_table
        i. On inactivity.start_dt = next_26.start_dt + INTERVAL '26 DAYS'
2. Create flag if
    a. Another 13 weeks inactivity (atleast 13-weeks)
    b. Else 'seller activated'

—---

```
Next_26 as
(
Select
R.seller_id,
Fm.migration_week,
sum(case when gmv>0 then 1 else 0 end) as reactivation_count
From rolling_13 r
Join first_migration fm
On rolling_13.seller_id = first_migration.seller_id
And rolling_13.retail_week = first_migration.migration_week + INTERVAL ‘26 WEEKS’
)
Select
Seller_id,
First_migration_year as inactive_year,
First_migration_week as inactive_week,
Case when reactivation >0 then ‘INACTIVITY BROKEN’ else ‘STILL INACTIVE’ end as inactive_flag
From next_26
```

Output →

```
Week 3 | min_migrated_week | 1200 GMV
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
```