Question: Extract the farmer's products that have prices above the market date's average product cost only for vendor having id 8.

```
4  SELECT *
5  FROM
6  (SELECT vendor_id,product_id,market_date,
7  original_price,
8  ROUND(AVG(original_price) OVER (PARTITION BY  market_date ORDER BY  market_date),2)  AS average_cost_product_by_market_date
9  FROM vendor_inventory ) AS temp
10 WHERE temp.vendor_id=8
11 AND temp.original_price> temp.average_cost_product_by_market_date
12 ORDER BY  temp.market_date,temp.original_price DESC
13
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_id	product_id	market_date	original_price	average_cost_product_by_market_date
8	7	2019-04-03	18.00	11.63
8	8	2019-04-03	18.00	11.63
8	7	2019-04-06	18.00	11.63
8	8	2019-04-06	18.00	11.63
8	7	2019-04-10	18.00	11.63
8	8	2019-04-10	18.00	11.63
8	7	2019-04-13	18.00	11.63
8	8	2019-04-13	18.00	11.63
8	7	2019-04-17	18.00	11.63
8	8	2019-04-17	18.00	11.63
8	7	2019-04-20	18.00	11.63
8	8	2019-04-20	18.00	11.63

Question: Count how many different products each vendor brought to market on each date and displays that count on each row.

```
4 •   SELECT vendor_id,market_date,product_id,
5     COUNT(product_id) OVER ( PARTITION BY market_date,vendor_id )
6     FROM vendor_inventory
7     order by  vendor_id,market_date ,original_price desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_id	market_date	product_id	COUNT(product_id) OVER (PARTITION BY market_date,vendor_id)
7	2019-06-12	4	1
7	2019-06-15	4	1
7	2019-06-19	4	1
7	2019-06-22	4	1
7	2019-06-26	4	1
7	2019-06-29	4	1
7	2019-07-03	1	4
7	2019-07-03	4	4
7	2019-07-03	2	4
7	2019-07-03	3	4
7	2019-07-06	1	4
7	2019-07-06	4	4
7	2019-07-06	2	4

Question: Calculate the running total of the cost of items purchased by each customer, sorted by the date and time and the product_id

```
4 •  SELECT customer_id, vendor_id,
5    market_date,
6    quantity*cost_to_customer_per_qty as price,
7    SUM( quantity*cost_to_customer_per_qty )
8    OVER (PARTITION BY customer_id ORDER BY market_date,transaction_time,product_id ROWS UNBOUNDED PRECEDING ) as
9    running_total_customer
10   FROM customer_purchases
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	vendor_id	market_date	price	running_total_customer
1	7	2020-09-30	2.0000	3416.8682
1	7	2020-09-30	1.5000	3418.3682
1	7	2020-09-30	8.5505	3426.9187
1	7	2020-10-07	7.0000	3433.9187
1	8	2020-10-07	36.0000	3469.9187
1	7	2020-10-07	14.0000	3483.9187
1	7	2020-10-10	17.5000	3501.4187
1	7	2020-10-10	3.5000	3504.9187
1	8	2020-10-10	26.0000	3530.9187
2	8	2019-04-06	6.5000	6.5000
2	7	2019-04-06	20.0000	26.5000
2	8	2019-04-10	18.0000	44.5000
2	8	2019-04-10	6.5000	51.0000

Question: Display each vendor's booth assignment for each market_date alongside their previous booth assignments.

```
4 •  SELECT  vendor_id,
5    market_date , booth_number,
6    LAG(booth_number) OVER(PARTITION by vendor_id ORDER BY market_date,vendor_id) as previos_booth_number
7    FROM vendor_booth_assignments
8
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_id	market_date	booth_number	previos_booth_number
1	2019-04-03	2	NULL
1	2019-04-06	2	2
1	2019-04-10	7	2
1	2019-04-13	2	7
1	2019-04-17	2	2
1	2019-04-20	2	2
1	2019-04-24	2	2
1	2019-04-27	2	2
1	2019-05-01	2	2
1	2019-05-04	2	2
1	2019-05-08	2	2
1	2019-05-11	2	2
1	2019-05-15	2	2
1	2019-05-18	2	2
1	2019-05-22	2	2
1	2019-05-25	2	2
1	2019-05-29	2	2

Question: On specific market date ,determine which vendors are

new or changing booths that day, so we can contact them and ensure setup goes smoothly.

Check it for date: 2019-04-10

```
5    SELECT *
6    FROM (
7    SELECT vendor_id,booth_number,market_date,
8    LAG(booth_number) OVER (PARTITION BY vendor_id  ORDER BY market_Date,vendor_id ) as previous_booth_number
9    FROM vendor_booth_assignments
10    ) as x
11   WHERE market_date="2019-04-10"
12   and  (x.booth_number <> x.previous_booth_number OR x.previous_booth_number IS NULL )
13
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_id	booth_number	market_date	previous_booth_number
1	7	2019-04-10	2
4	2	2019-04-10	7

Question: To get a profile of each customer's habits over time

```
3 •  SELECT customer_id,
4    MAX(market_date) as  last_purchase_date,
5    MIN(market_date) as first_purchase_date,
6     DATEDIFF(MAX(market_date),MIN(market_date)) as days_between_first_last_purchase,
7     COUNT(DISTINCT market_date) as count_of_purchase_dates,
8     SUM( quantity * cost_to_customer_per_qty) as total_amount_spent
9     FROM customer_purchases
10    GROUP BY customer_id
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	last_purchase_date	first_purchase_date	days_between_first_last_purchase	count_of_purchase_dates	total_amount_spent
1	2020-10-10	2019-04-06	553	107	3530.9187
2	2020-10-10	2019-04-06	553	117	4179.4529
3	2020-10-07	2019-04-03	553	112	3832.1575
4	2020-10-03	2019-04-03	549	115	3561.6286
5	2020-10-10	2019-04-03	556	113	3932.8271
6	2020-10-10	2019-04-03	556	95	3016.4655
7	2020-10-10	2019-04-03	556	100	2921.1743
8	2020-10-10	2019-04-06	553	101	3403.6797
9	2020-10-10	2019-04-03	556	92	3015.7341
10	2020-10-10	2019-04-03	556	96	2495.4089

Question: Write a query that gives us the days between each purchase a customer makes.

```
5  SELECT *,
6  IFNULL(DATEDIFF(market_date, previous_purchase_date),"First Purchase") as date_diff_between_purchases
7  FROM (
8  SELECT DISTINCT customer_id,
9  market_date,
10 IFNULL(LAG(market_date) OVER( PARTITION BY customer_id order by market_date,customer_id),0) as previous_purchase_date
11 FROM customer_purchases  ) as x
12 WHERE market_date!=previous_purchase_Date
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	market_date	previous_purchase_date	date_diff_between_purchases
1	2019-04-06	0	First Purchase
1	2019-04-13	2019-04-06	7
1	2019-04-17	2019-04-13	4
1	2019-04-20	2019-04-17	3
1	2019-04-24	2019-04-20	4
1	2019-04-27	2019-04-24	3
1	2019-05-01	2019-04-27	4
1	2019-05-04	2019-05-01	3
1	2019-05-08	2019-05-04	4
1	2019-05-15	2019-05-08	7

Question: today's date is March 31, 2019, and the marketing director of the farmer's market wants to give infrequent 10 customers an incentive to return to the market in April.

```
5  SELECT customer_id,
6  COUNT(DISTINCT market_date) as count_of_market_visit_in_march
7  FROM customer_purchases
8  WHERE DATEDIFF(market_date,"2019-03-31")  < 31
9  GROUP BY customer_id
10 order by count_of_market_visit_in_march
11 Limit 10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch ro

customer_id	count_of_market_visit_in_march
19	1
15	1
26	2
14	2
22	2
21	2
16	3
6	3
17	3
20	3