

Report: report_customer_median

Conditions:

- Customer name format as (x. xxx)
- Total amount spent format as (\$d,ddd.dd)
- Only customers bought more than 5 books
- Calculate median value using analytical function

```
WITH x AS (SELECT s.customer_id,
                  SUBSTR(first_name, 1, 1) || '. ' || last_name AS full_name,
                  total_amount,
                  PERCENTILE_DISC(0.5) WITHIN GROUP(ORDER BY total_amount)
                  OVER(PARTITION BY s.customer_id) AS median_value
FROM sales s
LEFT JOIN customer c ON c.customer_id = s.customer_id
)

SELECT customer_id,
       full_name,
       COUNT(customer_id) AS books_bought,
       TO_CHAR(SUM(total_amount), '$9,999.99') AS spent,
       median_value

FROM x
WHERE median_value >= 10

GROUP BY customer_id,
         full_name,
         median_value
HAVING COUNT(customer_id) > 5

ORDER BY 1 ;
```

Query Result x					
All Rows Fetched: 20 in 0.116 seconds					
	CUSTOMER_ID	FULL_NAME	BOOKS_BOUGHT	SPENT	MEDIAN_VALUE
1	6	S. KIM	7	\$78.72	10.87
2	8	L. LEOSTRA	8	\$105.84	12.49
3	12	T. JIE	7	\$83.04	12.31
4	13	A. WEAVER	8	\$104.98	13.28
5	18	S. OLSEN	8	\$95.89	11.09
6	23	A. NOTRE	10	\$122.70	11.92
7	24	K. BACH	6	\$74.60	13.62
8	26	O. FABIAN	7	\$91.60	12.31
9	27	G. JOHNSON	9	\$110.35	12.07
10	29	A. GORR	11	\$133.32	13.17

Report: report_last_purchase

Conditions:

- Show previous purchase using analytical functions
- Show total of the last 3 purchases using analytical functions
- Date period 2020

```
SELECT sale_id,
       sales_date,
       customer_id,
       total_amount AS current_purchase,
       NVL(LAG(total_amount, 1) OVER(PARTITION BY customer_id
                                     ORDER BY sale_id), 0) AS previous_purchase,
       NVL(SUM(total_amount) OVER(PARTITION BY customer_id
                                  ORDER BY sale_id
                                  ROWS BETWEEN 3 PRECEDING AND 1 PRECEDING), 0) AS last_three_purchases
FROM sales
WHERE sales_date >= TO_DATE('01-JAN-2020', 'dd-mm-yyyy')
      AND sales_date < TO_DATE('01-JAN-2021', 'dd-mm-yyyy')
ORDER BY 1 ;
```

Query Result x

SQL | All Rows Fetched: 124 in 0.235 seconds

SALE_ID	SALES_DATE	CUSTOMER_ID	CURRENT_PURCHASE	PREVIOUS_PURCHASE	LAST_THREE_PURCHASES
59	59 01-DEC-20	37	13.84	18.34	33.54
60	60 01-DEC-20	50	16.95	0	0
61	61 07-DEC-20	25	8.46	13.17	18.58
62	62 07-DEC-20	33	8.59	7.75	7.75
63	63 07-DEC-20	31	10.87	0	0
64	64 07-DEC-20	8	14.66	14.75	33.08
65	65 07-DEC-20	34	5.54	8.67	20.46
66	66 07-DEC-20	46	11.9	16.89	16.89
67	67 07-DEC-20	50	17.16	16.95	16.95
68	68 07-DEC-20	9	12.98	0	0
69	69 07-DEC-20	23	10.87	8.31	20.23
70	70 07-DEC-20	2	12.31	0	0
71	71 07-DEC-20	32	18.75	7.5	22.41
72	72 07-DEC-20	31	4.28	10.87	10.87
73	73 10-DEC-20	26	15.02	18.33	18.33
74	74 10-DEC-20	35	13.48	7.75	7.75
75	75 10-DEC-20	50	14.87	17.16	34.11

Report: report_top_books_sold

Conditions:

- Find ten books with the highest sales
- Find information of the books joining AUTHOR table

```
SELECT s.book_id,  
       book_name,  
       translated,  
       first_name || ' ' || last_name AS author_name,  
       COUNT(s.book_id) AS sold  
  
FROM sales s  
LEFT JOIN book b ON b.book_id = s.book_id  
LEFT JOIN author a ON a.author_id = b.author_id  
  
GROUP BY s.book_id,  
         book_name,  
         translated,  
         first_name || ' ' || last_name  
  
ORDER BY sold DESC  
FETCH NEXT 10 ROWS ONLY ;
```

Query Result x

   SQL | All Rows Fetched: 10 in 0.129 seconds

	BOOK_ID	BOOK_NAME	TRANSLATED	AUTHOR_NAME	SOLD
1	87	DANDELION WINE	SPANISH	RAY BRADBURY	7
2	13	SOLARIS	ENGLISH	STANISLAW LEM	6
3	151	THE OLD MAN AND THE SEA	FRENCH	ERNEST HEMINGWAY	5
4	41	THE SILMARILLION	RUSSIAN	JOHN TOLKIEN	5
5	153	FOR WHOM THE BELL TOLLS	FRENCH	ERNEST HEMINGWAY	4
6	33	FOR WHOM THE BELL TOLLS	ENGLISH	ERNEST HEMINGWAY	4
7	117	OLIVER TWIST	SPANISH	CHARLES DICKENS	4
8	76	THE GREAT GATSBY	RUSSIAN	SCOTT FITZGERALD	4
9	129	THE DARK TOWER: WOLFS OF THE CALLA	FRENCH	STEPHEN KING	4
10	70	THE STAND	RUSSIAN	STEPHEN KING	4