Contents

[1. Task 1 2](#_Toc498465319)

[Task 2](#_Toc498465320)

[Script 2](#_Toc498465321)

[Result 3](#_Toc498465322)

[Conclusions 3](#_Toc498465323)

[2. Task 2 4](#_Toc498465324)

[Task 4](#_Toc498465325)

[Script 4](#_Toc498465326)

[Result 5](#_Toc498465327)

[Conclusions 5](#_Toc498465328)

[3. Task 3 6](#_Toc498465329)

[Task 6](#_Toc498465330)

[Script 6](#_Toc498465331)

[Result 6](#_Toc498465332)

[Conclusions 6](#_Toc498465333)

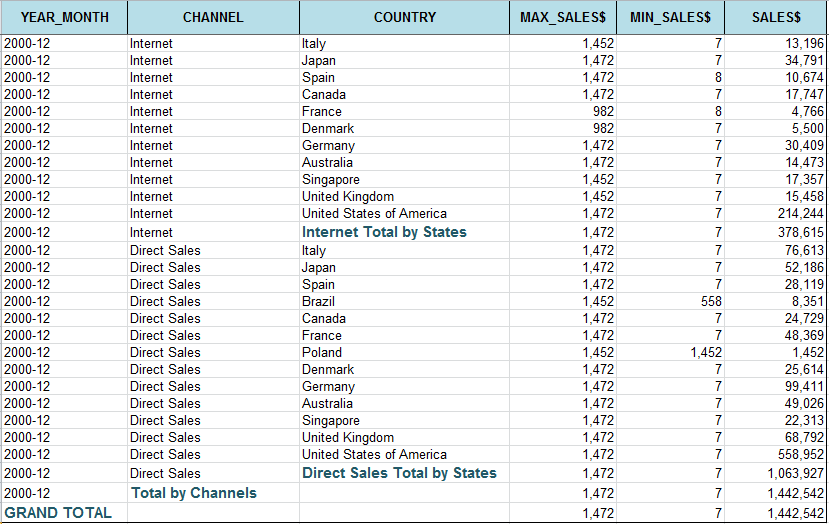
# Task 1

### Task

Analyze max, min and summary sales in the report below by:

* December 2000;
* Two channels: Internet and Direct Sales;
* All States.

Build the query using Oracle SQL for Aggregation and design the same report (export results to Excel).



### Script

SELECT DECODE(GROUPING(calendar\_month\_desc),1, 'GRAND TOTAL', calendar\_month\_desc) AS YEAR\_MONTH,

CASE

WHEN GROUPING\_id(calendar\_month\_desc, channel\_desc)=1

THEN 'Total by Channels'

WHEN sh.channels.channel\_desc IS NULL

THEN' '

ELSE sh.channels.channel\_desc

END AS Channel,

CASE

WHEN grouping\_id(sh.channels.channel\_desc,sh.countries.country\_name)=1

THEN sh.channels.channel\_desc

|| ' Total By States'

WHEN sh.countries.country\_name IS NULL

THEN' '

ELSE sh.countries.country\_name

END AS Country,

ROUND(MAX(amount\_sold)) AS MAX\_SALES$,

ROUND(MIN(amount\_sold)) AS MIN\_SALES$,

ROUND(SUM(amount\_sold)) AS SALES$

FROM sh.sales

LEFT JOIN sh.channels

ON sh.sales.channel\_id=sh.channels.channel\_id

LEFT JOIN sh.times

ON sh.sales.time\_id=sh.times.time\_id

LEFT JOIN sh.customers

ON sh.sales.cust\_id=sh.customers.cust\_id

LEFT JOIN sh.countries

ON sh.customers.country\_id=sh.countries.country\_id

WHERE calendar\_month\_desc ='2000-12'

AND (channel\_desc = 'Internet'

OR channel\_desc = 'Direct Sales')

GROUP BY ROLLUP(calendar\_month\_desc, channel\_desc, country\_name);

### Result



### Conclusions

CASE

WHEN grouping\_id(sh.channels.channel\_desc,sh.countries.country\_name)=1

THEN sh.channels.channel\_desc

|| ' Total By States'

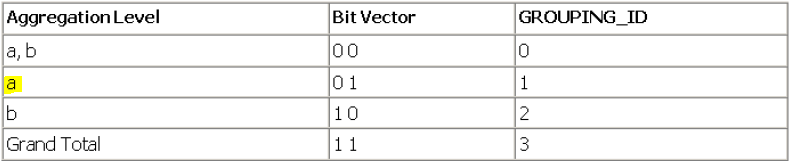
WHEN sh.countries.country\_name IS NULL

THEN' '

ELSE sh.countries.country\_name

END

1. Concatenation allows dynamically insert required cannel\_desc.
2. Grouping\_is is used: a is our situation, when country\_name is null because of rollup grouping\_id is 1.

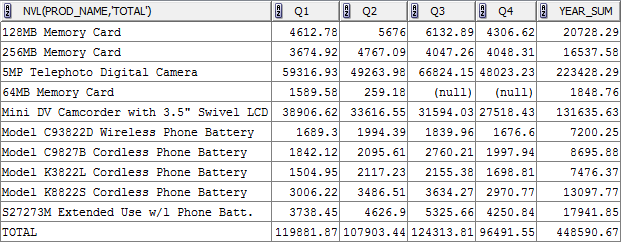


1. Here also NULL replacement is implemented, but it’s **not necessary** , because when we export in Excel it will be shown as blank space.
2. Groupings and grouping\_id’s of different levels were used.
3. GROUP BY ROLLAP was used.

# Task 2

### Task

Напишите запрос для формирования отчёта о продажах в Азии всех продуктов из категории Photo за 2000 год по кварталам. Вычислите итоговые суммы (TOTAL и YEAR\_SUM).



### Script

WITH quarter AS

( SELECT DISTINCT sh.products.prod\_name AS prod\_name,

NTH\_VALUE(SUM(AMOUNT\_SOLD),1) OVER (PARTITION BY prod\_name ORDER BY sh.times .calendar\_quarter\_number ASC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Q1,

NTH\_VALUE(SUM(AMOUNT\_SOLD),2) OVER (PARTITION BY prod\_name ORDER BY sh.times .calendar\_quarter\_number ASC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Q2,

NTH\_VALUE(SUM(AMOUNT\_SOLD),3) OVER (PARTITION BY prod\_name ORDER BY sh.times .calendar\_quarter\_number ASC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Q3,

NTH\_VALUE(SUM(AMOUNT\_SOLD),4) OVER (PARTITION BY prod\_name ORDER BY sh.times .calendar\_quarter\_number ASC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Q4,

SUM(SUM(AMOUNT\_SOLD)) OVER (PARTITION BY prod\_name) AS YEAR\_SUM

FROM sh.sales

LEFT JOIN sh.products

ON sh.sales.prod\_id=sh.products.prod\_id

LEFT JOIN sh.times

ON sh.sales.time\_id=sh.times.time\_id

LEFT JOIN sh.channels

ON sh.sales.channel\_id=sh.channels.channel\_id

LEFT JOIN sh.customers

ON sh.sales.cust\_id=sh.customers.cust\_id

LEFT JOIN sh.countries

ON sh.customers.country\_id =sh.countries.country\_id

WHERE sh.times.calendar\_year ='2000'

AND country\_region ='Asia'

AND prod\_category ='Photo'

GROUP BY sh.products.prod\_name,

sh.times .calendar\_quarter\_number

)

SELECT NVL(sh.products.prod\_name,'TOTAL') AS Prod\_name,

SUM(Q1) AS Q1,

SUM(Q2) AS Q2,

SUM(Q3) AS Q3,

SUM(Q4) AS Q4,

SUM(year\_sum)

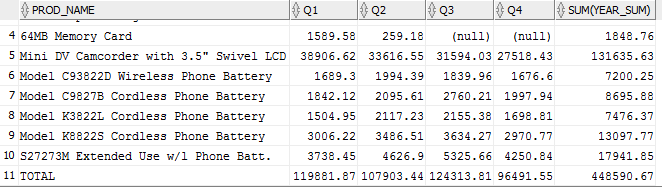
FROM quarter q

LEFT JOIN sh.products

ON q.prod\_name=sh.products.prod\_name

GROUP BY rollup(sh.products.prod\_name) ;

### Result



### Conclusions

1. NTH\_VALUE function was used:

NTH\_VALUE (SUM(AMOUNT\_SOLD),4) OVER (PARTITION BY prod\_name

ORDER BY sh.times .calendar\_quarter\_number ASC

RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)

NTH\_VALUE lets you find a **specific position in the analytic window** such as 2nd, 3rd, or 4th value

We partition our table by prod\_name’s and then ordered them by quarters:

|  |  |  |
| --- | --- | --- |
| Prod 1 | Q1 | 1st value |
| Q1 | 1st value |
| Q1 | 1st value |
| Q2 | 2nd value |
| Q2 | 2nd value |
| Prod 2 | Q1 | 1st value |
| Q2 | 2nd value |
| … |  |  |

RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING – All rows are included in the window, regardless of the current row.

# Task 3

### Task

Show how many products of each model were sold in 2000 and 2001 in Asia and Europe.

### Script

SELECT \*

FROM

(SELECT prod\_name,

COUNTRY\_REGION,

times.calendar\_year,

QUANTITY\_SOLD

FROM sh.sales

LEFT JOIN sh.products

ON sh.sales.prod\_id=sh.products.prod\_id

LEFT JOIN sh.times

ON sh.sales.time\_id=sh.times.time\_id

LEFT JOIN sh.customers

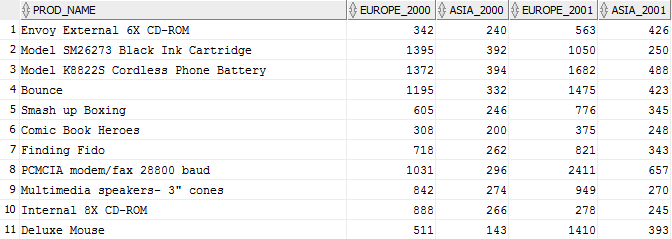
ON sh.sales.cust\_id=sh.customers.cust\_id

LEFT JOIN sh.countries

ON sh.customers.country\_id =sh.countries.country\_id

) PIVOT(SUM(QUANTITY\_SOLD) FOR (country\_region,calendar\_year) IN (('Europe','2000')AS Europe\_2000, ('Asia','2000')AS Asia\_2000, ('Europe',2001)AS Europe\_2001, ('Asia',2001)AS Asia\_2001 ) ) ;

### Result



### Conclusions

Pivot – which attribute we insert

For – in which columns/