|  |
| --- |
| , RD Dep. |
| **Introduction to OLAP** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial Version | Mikita Kandratsiuk | 14-Nov-2017 |  |  |
| 2.0 | Report overview | Hanna Klimovich | 15-NOV-2017 |  |  |

Contents

[1. Frameworks 3](#_Toc498425030)

[2. Presentation 3](#_Toc498425031)

[3. SQL for analysis 3](#_Toc498425032)

# 

# Presentation

Presentations about methodologies.

|  |  |  |
| --- | --- | --- |
| Aksana Kuratnik |  | Kanban |
| Aliaksandr Labayeu |  |  |
| Vitaliya Adamchuk |  |  |
|  |  |  |
| Saida Melikava |  | Scrum |
| Valeryia Lupanava |  |  |
|  |  |  |
| Alina Makarets |  | Agile |
| Arina Marchenko |  |  |
| Hanna Takushevich |  |  |
|  |  |  |
| Mikita Tur |  | Other |
| Olga Hilko |  |  |
|  |  |  |
| Gleb Kulvanovski |  | Waterfall |
| Anastasiya Khilko |  |  |
| Hanna Hul |  |  |

# SQL for analysis

1. Topic 2. Task 2 (please, make an example of usage GROUPING\_ID).

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).

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);



CASE

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

THEN sh.channels.channel\_desc

|| ' Total By States'

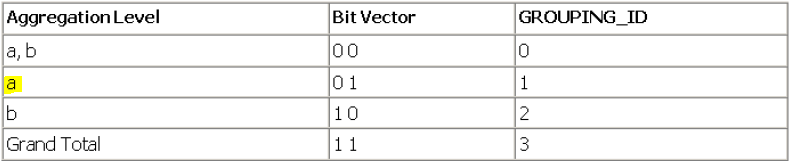
WHEN sh.countries.country\_name IS NULL

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.
4. Topic 3. Task 2.

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).

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,

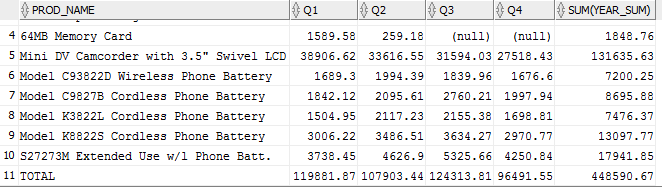
SUM(year\_sum)

FROM quarter q

LEFT JOIN sh.products

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

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



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.

1. Your personal example of PIVOT

CREATE TABLE SALES\_VIEW AS

SELECT PROD\_NAME PRODUCT,

COUNTRY\_NAME COUNTRY,

CHANNEL\_ID CHANNEL,

SUBSTR(CALENDAR\_QUARTER\_DESC, 6,2) QUARTER,

SUM(AMOUNT\_SOLD) AMOUNT\_SOLD,

SUM(QUANTITY\_SOLD) QUANTITY\_SOLD

FROM sh.SALES,

sh.TIMES,

sh.CUSTOMERS,

sh.COUNTRIES,

sh.PRODUCTS

WHERE sh.SALES.TIME\_ID = sh.TIMES.TIME\_ID

AND sh.SALES.PROD\_ID = sh.PRODUCTS.PROD\_ID

AND sh.SALES.CUST\_ID = sh.CUSTOMERS.CUST\_ID

AND sh.CUSTOMERS.COUNTRY\_ID = sh.COUNTRIES.COUNTRY\_ID

GROUP BY PROD\_NAME,

COUNTRY\_NAME,

CHANNEL\_ID,

SUBSTR(CALENDAR\_QUARTER\_DESC, 6, 2) ;

SELECT \*

FROM

(SELECT PRODUCT, CHANNEL, AMOUNT\_SOLD FROM SALES\_VIEW

) PIVOT (SUM(AMOUNT\_SOLD) FOR CHANNEL IN (2 AS PARTNERS, 3 AS DIRECT\_SALES, 4 AS INTERNET\_SALES, 9 AS TELESALES))

ORDER BY PRODUCT;