## Task 01: Create Ad Hoc SQL FIRST\_VALUE, LAST\_VALUE

Task goal: to identify tariffs that bring the highest and lowest profit by quarter from 2004 to 2008.

SELECT "YEAR" , "Quarter", tariff\_type, tariff\_name , tariff\_code , count\_tr , profit

FROM (SELECT "YEAR" , "Quarter" , tariff\_type , tariff\_name

, tariff\_code , count\_tr , profit ,

LAST\_VALUE (

profit

)

OVER ( PARTITION BY "YEAR", "Quarter"

ORDER BY profit

ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING )

AS lv

, FIRST\_VALUE (

profit

)

OVER ( PARTITION BY "YEAR", "Quarter"

ORDER BY profit

ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING )

AS fv

FROM ( SELECT TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' )

"YEAR"

, TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' )

"Quarter"

, tariff\_type

, tariff\_name

, tariff\_code

, COUNT ( payment\_sum / currency\_to\_dollar ) count\_tr

, ROUND ( SUM ( ( payment\_sum / currency\_to\_dollar ) \* tariff\_payment\_sum / 100 )

, 2 )

profit

FROM u\_dw\_ext\_references.tmp\_transactions\_info

WHERE tariff\_type IN ('Local Transfer', 'International Transfer')

AND event\_dt >= TRUNC ( TO\_DATE ( '01-JAN-2004'

, 'dd-mon-yyyy' )

, 'month' )

AND event\_dt <= (TRUNC ( TO\_DATE ( ADD\_MONTHS ( TO\_DATE ( '01-JAN-2008'

, 'dd-mon-yyyy' )

, 1 ) )

, 'month' )

- 1)

GROUP BY tariff\_type

, (tariff\_code, tariff\_name)

, TRUNC ( event\_dt

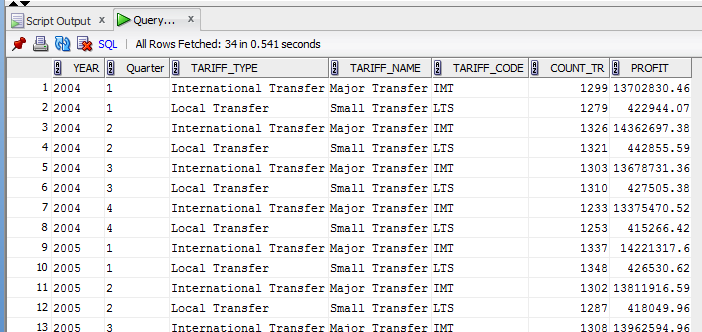
, 'YYYY' )

, TRUNC ( event\_dt

, 'Q' )))

WHERE profit = lv OR profit = fv

ORDER BY 1 , 2 , 3 , 7;



## Task 02: Create Ad Hoc SQL RANK, DENSE\_RANK, ROWNUM

Task goal: to identify the top 5 countries in which the number of operations "deposit" was the highest by quarter in 2007 and 2008.

SELECT \*

FROM (SELECT "YEAR"

, "Quarter"

, cust\_rec\_cntr

, count\_tr

, DENSE\_RANK ( ) OVER (PARTITION BY "YEAR", "Quarter" ORDER BY count\_tr) RANK

FROM ( SELECT TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' )

"YEAR"

, TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' )

"Quarter"

, cust\_rec\_cntr

, COUNT ( transaction\_id ) count\_tr

FROM u\_dw\_ext\_references.tmp\_transactions\_info

WHERE operation\_name = 'DEPOSIT'

AND TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' ) IN ('2007', '2008')

GROUP BY cust\_rec\_cntr

, TRUNC ( event\_dt

, 'YYYY' )

, TRUNC ( event\_dt

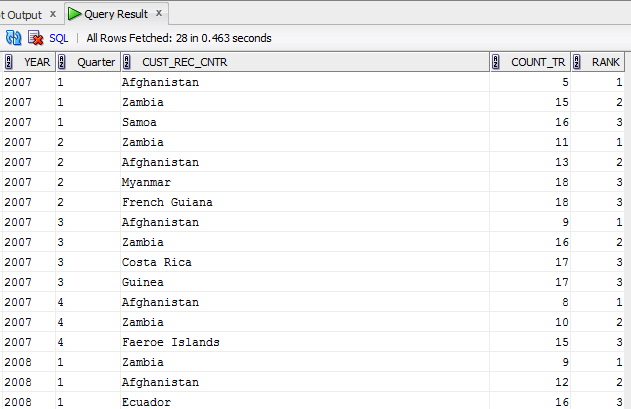
, 'Q' )))

WHERE RANK < 4

ORDER BY 1

, 2

, RANK;



Task goal: to identify the country with the maximum amount of payments for the operation a "deposit" on quarters of 2007 and 2008

SELECT "YEAR"

, "Quarter"

, cust\_rec\_cntr

, TO\_CHAR ( payment\_sum ) || '$' payment\_sum

FROM (SELECT "YEAR"

, "Quarter"

, cust\_rec\_cntr

, payment\_sum

, ROW\_NUMBER ( ) OVER (PARTITION BY "YEAR", "Quarter" ORDER BY payment\_sum) RANK

FROM ( SELECT TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' )

"YEAR"

, TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' )

"Quarter"

, cust\_rec\_cntr

, SUM ( payment\_sum ) payment\_sum

FROM u\_dw\_ext\_references.tmp\_transactions\_info

WHERE operation\_name = 'DEPOSIT'

AND TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' ) IN ('2007', '2008')

GROUP BY cust\_rec\_cntr

, TRUNC ( event\_dt

, 'YYYY' )

, TRUNC ( event\_dt

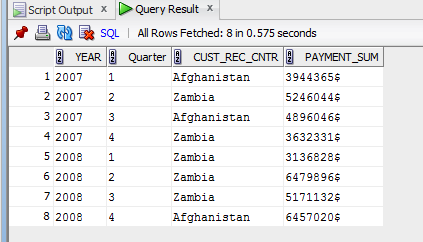
, 'Q' )))

WHERE RANK = 1

ORDER BY 1

, 2

, RANK;



Task goal: to determine the top 10 countries with the largest number of operations "deposit" in 2001 and 2002 by quarter.

SELECT \*

FROM (SELECT "YEAR"

, "Quarter"

, cust\_rec\_cntr

, count\_tr

, RANK ( ) OVER (ORDER BY count\_tr) RANK

FROM ( SELECT TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' )

"YEAR"

, TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' )

"Quarter"

, cust\_rec\_cntr

, COUNT ( transaction\_id ) count\_tr

FROM u\_dw\_ext\_references.tmp\_transactions\_info

WHERE operation\_name = 'DEPOSIT'

AND TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' ) IN ('2002', '2001')

GROUP BY cust\_rec\_cntr

, TRUNC ( event\_dt

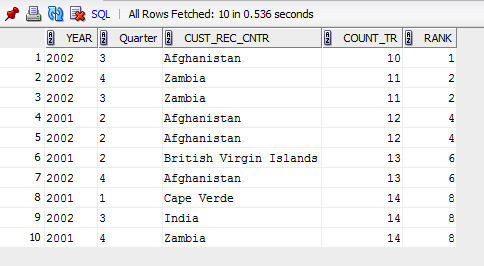
, 'YYYY' )

, TRUNC ( event\_dt

, 'Q' )))

WHERE RANK < 10

ORDER BY RANK;



## Task 03: Create Ad Hoc SQL AGGREAGATE FUNCS

SELECT DISTINCT

"YEAR"

, "Quarter"

, cust\_rec\_cntr

, payment\_sum

, CASE

WHEN payment\_sum = min\_payment\_sum THEN 'minimum payments' || ' ' || "YEAR" || '-' || "Quarter" || ' Q'

ELSE 'maximum payments' || ' ' || "YEAR" || '-' || "Quarter" || ' Q'

END

lv

FROM (SELECT "YEAR"

, "Quarter"

, cust\_rec\_cntr

, payment\_sum

, MIN ( payment\_sum ) OVER (PARTITION BY "YEAR", "Quarter") min\_payment\_sum

, MAX ( payment\_sum ) OVER (PARTITION BY "YEAR", "Quarter") max\_payment\_sum

FROM (SELECT TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' )

"YEAR"

, TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' )

"Quarter"

, cust\_rec\_cntr

, ROUND ( AVG ( payment\_sum ) OVER (PARTITION BY cust\_rec\_cntr) ) payment\_sum

FROM u\_dw\_ext\_references.tmp\_transactions\_info

WHERE operation\_name = 'DEPOSIT'

AND TO\_CHAR ( TRUNC ( event\_dt

, 'YYYY' )

, 'YYYY' ) IN ('2012')

AND TO\_CHAR ( TRUNC ( event\_dt

, 'Q' )

, 'Q' ) IN ('1')))

WHERE payment\_sum = min\_payment\_sum

OR payment\_sum = max\_payment\_sum

ORDER BY 1

, 2

, payment\_sum;

