

RDB&SQL Session-8 (Window Functions-1)

SQL Session-8 (Window Functions-1)
Training Clarusway
Pear Deck - January 23, 2023 at 7:57PM

Part 1 - Summary

Use this space to summarize your thoughts on the lesson

Part 2 - Responses

Slide 1



CLARUSWAY®
data to knowledge transformed

Use this space to take notes:

Slide 2



Window Functions



CLARUSWAY®
BORN TO REINVENT YOURSELF.

Use this space to take notes:

Slide 3

Your Response

I've completed the pre-class content?

True **False**

Pear Deck

Students choose an option

Pear Deck Interactive Slide
Do not refresh this tab

You Chose

- **False**

Other Choices

- True

Use this space to take notes:

Slide 4

Table of Contents ➤

- ▶ Window Functions (WF) vs. GROUP BY
- ▶ Types of WF
- ▶ WF Syntax and Keywords
- ▶ Window Frames
- ▶ How to Apply WF

CLARUSWAY®
WAY TO REINVENT YOURSELF.



Use this space to take notes:

Slide 5



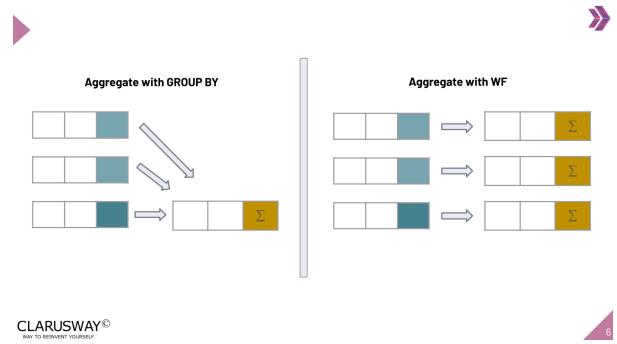
Window Functions (WF) vs. GROUP BY



CLARUSWAY®
WAY TO REINVENT YOURSELF.

Use this space to take notes:

Slide 6



Use this space to take notes:

Slide 7

▶ WF vs. GROUP BY

	GROUP BY	Window Functions
Distinct	necessity	optional
Aggregating	necessity	optional
Ordering	invalid	valid
Performance	slower	faster
Dependency on Selected Fields	dependent	independent

CLARUSWAY®
way to knowledge management

Use this space to take notes:

Slide 8

Query Time



Question: Write a query that shows the total stock amount of each product in the stock table.

(Use both of Group by and WF)

Expected Output:

#	product_id	Total Stock
1	1	65
2	2	45
3	3	34
4	4	36
5	5	26
6	6	38
7	7	28
8	8	13
9	9	51
10	10	49
11	11	59
12	12	48
*	*	*

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 9

Query Time For You



Question: Write a query that returns average product prices of brands.

(Use both of Group by and WF)

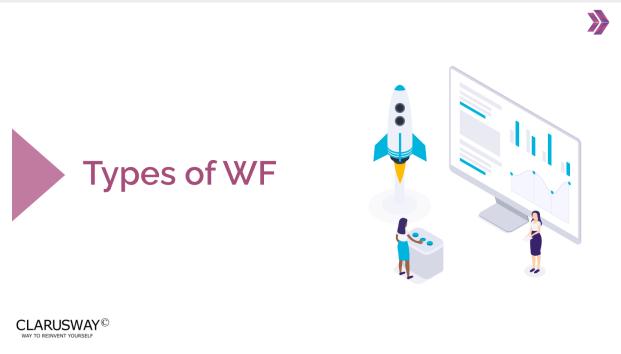
Expected Output:

#	brand_id	avg. price
1	1	1047.642195
2	2	527.851675
3	3	583.450000
4	4	103.440000
5	5	459.979565
6	6	193.500000
7	7	150.720256
8	8	124.616666
9	9	139.325333
10	10	534.462142
11	11	463.968571
12	12	257.387142
*	*	519.879555

CLARUSWAY®
WAY TO REINVENT YOURSELF

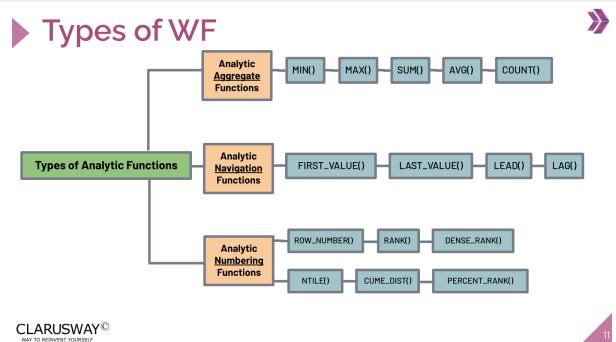
Use this space to take notes:

Slide 10



Use this space to take notes:

Slide 11



Use this space to take notes:

Slide 12

WF Syntax and Keywords



CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 13

Syntax and Keywords

```
SELECT {columns},  
       FUNCTIONS OVER (PARTITION BY ... ORDER BY ... WINDOW/FRAME)  
FROM table1;
```

id	date	time
1	2019-07-05	22
1	2019-04-15	26
2	2019-02-06	28
1	2019-01-02	30
2	2019-08-30	20
2	2019-03-09	22

```
SELECT *,  
       AVG(time) OVER (PARTITION BY id ORDER BY date  
                         ROWS BETWEEN 1 PRECEDING AND CURRENT ROW  
                   ) as avg_time  
FROM time_of_sales
```

Use this space to take notes:

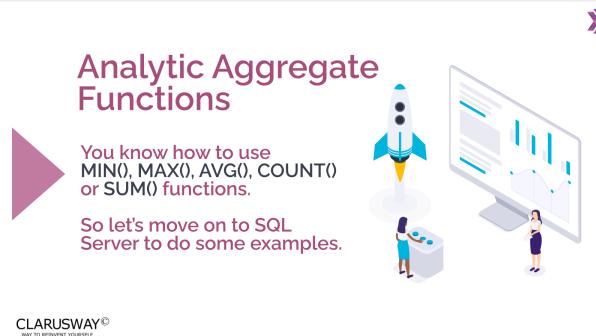
15

Slide 14



Use this space to take notes:

Slide 15



Use this space to take notes:

Slide 16

Query Time

What is the cheapest product price for each category?

Expected Output:

	category_id	cheapest_by_cat
1	1	3.00
2	4	1.00
3	5	22.99
4	6	29.99
5	7	81.99
6	8	499.95
7	9	55.95
8	10	232.99
9	11	33.99
10	13	49.99
11	14	39.99

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 17

Query Time

How many different product in the product table?

Expected Output:

	num_of_product
1	520

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 18

Query Time

How many different product in the order_item table?

Expected Output:

Results		Messages
num_of_product		
1	307	

1 DataSc... SampleData! 00:00:00 | 1 rows

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 19

Query Time

Write a query that returns how many products are in each order?

Expected Output:

Results			Messages
order_id cnt_product			
1	1	8	
2	2	3	
3	3	2	
4	4	2	
5	5	4	
6	6	8	
7	7	4	
8	8	3	
9	9	2	
10	10	1	
11	11	5	
12	12	3	
13	13	2	

1 DataSc... SampleData! 00:00:00 | 13 rows

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 20

Query Time



Write a query that returns the number of products in each category of brands.

Expected Output:

	category_id	brand_id	num_of_prod
1	1	1	15
2	1	3	10
3	1	15	8
4	1	22	6
5	1	32	1
6	4	1	22
7	4	2	41
8	4	3	5
9	4	4	20
10	4	6	38
11	4	8	15
12	4	9	13
13	4	10	14
14	4	11	14

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 21



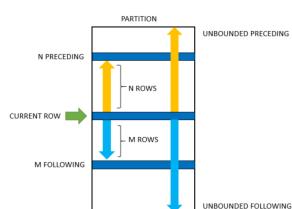
Window Frames



CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 22



★ Default: UNBOUNDED PRECEDING AND CURRENT ROW

CLARUSWAY®
WAY TO REINVENT YOURSELF.

22

Use this space to take notes:

Slide 23

```
SELECT *,  
AVG(time) OVER (PARTITION BY id ORDER BY date  
ROWS BETWEEN 1 PRECEDING AND CURRENT ROW  
) as avg_time  
FROM time_of_sales
```



CLARUSWAY®
WAY TO REINVENT YOURSELF.

23

Use this space to take notes:

Slide 24

Analytic Navigation Functions



CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 25

▶ First_Value Function

```
SELECT A.customer_id, A.first_name, B.order_date,  
       first_value (order_date) OVER (ORDER BY B.ORDER_DATE) first_date  
FROM   sale.customers A, sale.orders B  
WHERE  A.customer_id = B.customer_id
```

#	customer_id	first_name	order_date	first_date
1	259	Selma	2018-01-01	2018-01-01
2	1212	Jamie	2018-01-01	2018-01-01
3	523	James	2018-01-01	2018-01-01
4	175	Lloyd	2018-01-03	2018-01-01
5	1324	Ashleigh	2018-01-03	2018-01-01
6	1204	Tracey	2018-01-04	2018-01-01
7	224	John	2018-01-04	2018-01-01
8	94	Dick	2018-01-04	2018-01-01
9	60	Sue	2018-01-05	2018-01-01
10	442	James	2018-01-05	2018-01-01
11	1326	Laverne	2018-01-05	2018-01-01
12	91	Nicole	2018-01-06	2018-01-01
13	873	Marina	2018-01-08	2018-01-01
14	450	John	2018-01-09	2018-01-01
15	258	Tam	2018-01-09	2018-01-01

25

Use this space to take notes:

Slide 26

► First_Value Function

The diagram illustrates the use of the `First_Value` function across three stages of data processing:

- Stage 1:** A list of 15 order dates from 2018-01-01 to 2020-04-17.
- Stage 2:** The same data ordered by ascending date, labeled "ORDER ASC".
- Stage 3:** The data grouped by month, showing the first order date for each month, labeled "First Values".

28

Use this space to take notes:

Slide 27

Query Time

Write a query that returns first order date by month.

The diagram shows the expected output of a query that returns the first order date by month. The output is a table with the following structure:

	Year	Month	first_order_date
1	2018	1	2018-01-01
2	2018	2	2018-02-01
3	2018	3	2018-03-01
4	2018	4	2018-04-02
5	2018	5	2018-05-01
6	2018	6	2018-06-01
7	2018	7	2018-07-01
8	2018	8	2018-08-01
9	2018	9	2018-09-01
10	2018	10	2018-10-01
11	2018	11	2018-11-02

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 28

Query Time

Write a query that returns customers and their most valuable order with total amount of it.

Expected Output:

	customer_id	mv_order	mvorder	net_price
1	1	1555	1038.5370	
2	2	692	1470.8261	
3	3	1612	6763.3454	
4	4	1556	1420.1128	
5	5	264	1547.4950	
6	6	1611	2875.3547	
7	7	104	369.9725	
8	8	512	271.4414	
9	9	1593	5909.3979	
10	10	154	6561.3414	
11	11	1074	1880.8931	
12	12	1576	1420.0000	
13	13	238	2904.1041	
14	14	1572	3467.4380	

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 29

▶ Last_Value Function

```
SELECT A.customer_id, A.first_name, B.order_date,  
       last_value(order_date) OVER (ORDER BY B.ORDER_DATE DESC) last_date  
FROM sale.customer A, sale.orders B  
WHERE A.customer_id = B.customer_id
```

	customer_id	first_name	order_date	last_date
1	139	Ernest	2020-12-28	2020-12-28
2	135	Bruno	2020-11-18	2020-11-18
3	1	Davne	2020-11-18	2020-11-18
4	3	Teddy	2020-10-21	2020-10-21
5	6	Cyril	2020-09-01	2020-09-06
6	15	Susanne	2020-08-25	2020-08-25
7	19	Melissa	2020-08-23	2020-08-23
8	53	Trinidad	2020-07-12	2020-07-12
9	33	Yuki	2020-07-11	2020-07-11
10	119	Hilma	2020-07-10	2020-07-10
11	123	Jeni	2020-07-01	2020-07-01
12	7	William	2020-06-17	2020-06-17
13	35	Carrie	2020-04-30	2020-04-30
14	74	Eduardo	2020-04-30	2020-04-30
15	90	Daniel	2020-04-29	2020-04-29

29

Use this space to take notes:

Slide 30

► Last_Value Function

The figure consists of three side-by-side screenshots of a database query results window. The first screenshot shows a table with 15 rows of 'order_date' values from 2018-12-09 to 2020-04-21. A yellow arrow labeled 'ORDER DESC' points to the second screenshot. The second screenshot shows the same data with an 'OVER' clause added to the 'ORDER BY' clause, indicating a window function. A yellow arrow labeled 'Last Values' points to the third screenshot. The third screenshot shows the final result with an alias 'last_date' applied to the last value of each window, which is 2020-12-28.

order_date
2018-12-09
2020-04-18
2020-11-18
2020-04-09
2019-02-05
2019-08-21
2020-04-06
2020-10-21
2020-03-27
2019-11-21
2018-04-18
2019-02-07
2019-06-10
2018-11-24
2020-04-17

order_date
2020-12-28
2020-11-28
2020-11-18
2020-10-21
2020-09-06
2020-08-25
2020-08-23
2020-07-12
2020-07-11
2020-07-01
2020-06-17
2020-04-30
2020-04-29

last_date
2020-12-28
2020-11-28
2020-11-18
2020-10-21
2020-09-06
2020-08-25
2020-08-23
2020-07-12
2020-07-11
2020-07-10
2020-07-01
2020-06-17
2020-04-30
2020-04-29

Use this space to take notes:

Slide 31

► Last_Value Function

The figure shows a screenshot of a database query results window. At the top, there is a SQL query:SELECT B.customer_id, A.first_name, B.order_date,
 last_value(order_date) OVER (ORDER BY B.order_date DESC ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) last_date
FROM sale.customer A, sale.orders B
WHERE A.customer_id = B.customer_id;Below the query is a table with 15 rows of data. The columns are customer_id, first_name, order_date, and last_date. The last_date column contains the value 2018-01-01 for all rows, indicating that the last value of the order_date window is constant across all rows.

customer_id	first_name	order_date	last_date
1 136	Ernest	2020-12-28	2018-01-01
2 135	Pasquale	2020-11-28	2018-01-01
3 3	Bruno	2020-10-21	2018-01-01
4 3	Freddy	2020-10-21	2018-01-01
5 6	Cyril	2020-09-06	2018-01-01
6 15	Siohan	2020-08-25	2018-01-01
7 10	Marianna	2020-08-25	2018-01-01
8 53	Trinidade	2020-07-12	2018-01-01
9 33	Yuki	2020-07-11	2018-01-01
10 119	Armando	2020-07-10	2018-01-01
11 123	Jeanne	2020-06-17	2018-01-01
12 7	William	2020-06-17	2018-01-01
13 55	Carme	2020-04-30	2018-01-01
14 74	Nathalie	2020-04-30	2018-01-01
15 90	Daniel	2020-04-29	2018-01-01

Use this space to take notes:

Slide 32

Query Time



Write a query that returns first order date by month. (Use Last_Value)

Expected Output:

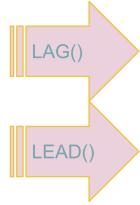
CLARUSWAY®
WAY TO REINVENT YOURSELF

	Year	Month	first_order_date
1	2018	1	2018-01-01
2	2018	2	2018-02-01
3	2018	3	2018-03-01
4	2018	4	2018-04-02
5	2018	5	2018-05-01
6	2018	6	2018-06-01
7	2018	7	2018-07-01
8	2018	8	2018-08-01
9	2018	9	2018-09-01
10	2018	10	2018-10-01
11	2018	11	2018-11-02
12	2018	12	2018-12-03

Use this space to take notes:

Slide 33

▶ Lag() & Lead() Functions



Returns the value in previous rows for each row of sorted column values.

Returns the value in next rows for each row of sorted column values.

33

Use this space to take notes:

Slide 34

▶ Lag() Function



```
SELECT order_date,  
       LAG(order_date) OVER (ORDER BY order_date) previous_w_LAG  
FROM sale.orders
```

#	order_date
1	2018-01-01
2	2018-01-01
3	2018-01-02
4	2018-01-03
5	2018-01-03
6	2018-01-04
7	2018-01-04
8	2018-01-04
9	2018-01-05
10	2018-01-05
11	2018-01-05
12	2018-01-06
13	2018-01-06
14	2018-01-08
15	2018-01-09



#	previous_w_LAG
1	NULL
2	2018-01-01
3	2018-01-02
4	2018-01-02
5	2018-01-03
6	2018-01-03
7	2018-01-04
8	2018-01-04
9	2018-01-04
10	2018-01-05
11	2018-01-05
12	2018-01-05
13	2018-01-06
14	2018-01-08
15	2018-01-09

34

Use this space to take notes:

Slide 35

▶ Lead() Function



```
SELECT order_date,  
       LEAD(order_date, 2) OVER (ORDER BY order_date) next_second_w_LEAD  
FROM sale.orders
```

#	order_date
1	2018-01-01
2	2018-01-01
3	2018-01-02
4	2018-01-03
5	2018-01-03
6	2018-01-04
7	2018-01-04
8	2018-01-04
9	2018-01-05
10	2018-01-05
11	2018-01-05
12	2018-01-06
13	2018-01-06
14	2018-01-08
15	2018-01-09



#	next_second_w_LEAD
1	2018-01-02
2	2018-01-03
3	2018-01-03
4	2018-01-04
5	2018-01-04
6	2018-01-04
7	2018-01-05
8	2018-01-05
9	2018-01-05
10	2018-01-06
11	2018-01-06
12	2018-01-09
13	2018-01-09
14	2018-01-12
15	2018-01-12

35

Use this space to take notes:

Slide 36

Query Time

Write a query that returns the order date of the one previous sale of each staff (use the LAG function)

#	order_id	staff_id	first_name	last_name	order_date	previous_order_date
1	1	2	Charles	Cussona	2018-01-01	2018-01-01
2	9	2	Charles	Cussona	2018-01-05	2018-01-01
3	12	2	Charles	Cussona	2018-01-06	2018-01-05
4	19	2	Charles	Cussona	2018-01-14	2018-01-06
5	20	2	Charles	Cussona	2018-01-14	2018-01-14
6	22	2	Charles	Cussona	2018-01-16	2018-01-14
7	23	2	Charles	Cussona	2018-01-16	2018-01-16
8	52	2	Charles	Cussona	2018-02-03	2018-01-16
9	62	2	Charles	Cussona	2018-02-07	2018-02-03

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 37

Query Time For You

2. Write a query that returns the order date of the one next sale of each staff (use the LEAD function)

Expected Output:

#	order_id	staff_id	first_name	last_name	order_date	next_order_date
1	1	2	Charles	Cussona	2018-01-01	2018-01-05
2	9	2	Charles	Cussona	2018-01-05	2018-01-06
3	12	2	Charles	Cussona	2018-01-06	2018-01-14
4	19	2	Charles	Cussona	2018-01-14	2018-01-14
5	20	2	Charles	Cussona	2018-01-14	2018-01-16
6	22	2	Charles	Cussona	2018-01-16	2018-01-16
7	23	2	Charles	Cussona	2018-01-16	2018-02-03
8	52	2	Charles	Cussona	2018-02-03	2018-02-07
9	62	2	Charles	Cussona	2018-02-07	2018-03-12
10	72	2	Charles	Cussona	2018-02-12	2018-02-16
11	77	2	Charles	Cussona	2018-02-16	2018-02-25

CLARUSWAY®
WAY TO REINVENT YOURSELF

Use this space to take notes:

Slide 38

Your Response

Slide 38

Your Response

Is everything clear so far?

Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

Use this space to take notes:

Slide 39

Your Response

How well did you like this lesson?

Students, drag the icon!

Pear Deck Interactive Slide
Do not remove this bar

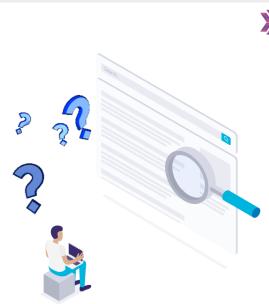
Use this space to take notes:

THANKS!

Any questions?

You can find me at:

- ▶ @adsum
- ▶ @owen



Use this space to take notes: