#56daysofSQL

Day 53
Window Functions
Part 3

Aditya Shenoy

#56daysofSQL

Value Window Functions

- LAG()
- LEAD()
- FIRST_VALUE
- LAST_VALUE

Aditya Shenoy

LAG()

- The **LAG()** window function returns the value for the row before the current row in a partition.
- If no row exists, null is returned.

General Syntax

```
LAG( expression )
OVER ( [ PARTITION BY expr_list ]
       [ ORDER BY order_list ]  )
```

#56daysofSQL

Use Case Example of LAG() :

The following example uses the **LAG** window function to show the quantity of records sold to the Tower Records customer with customer ID 8 and the dates that customer 8 purchased records:

```
SELECT cust_id, date, qty_sold, LAG(qty_sold,1) OVER
(order by cust_id, date) AS prev_qtysold
FROM sales
WHERE cust_id = 8
ORDER BY cust_id, date;
```

Output :

cust_id	date	qty_sold	prev_qtysold
8	1976-01-25	2	null
8	1981-02-04	5	2
8	1982-08-09	2	5
8	1983-02-12	1	2
8	1984-02-10	9	1

5 rows selected (0.331 seconds)

Since there is no purchase before 1976-01-25, the first value is **null**

LEAD()

- The **LEAD()** window function returns the value for the row after the current row in a partition.
- If no row exists, null is returned.

General Syntax

```
LEAD( expression )
OVER ( [ PARTITION BY expr_list ]
       [ ORDER BY order_list ]  )
```

#56daysofSQL

Use Case Example of LEAD():

The following example uses the **LEAD** window function to provide the commission for concert tickets with show ID 172 and the next commission for subsequent ticket sales:

```
SELECT show_id, date, commission,
LEAD(commission,1) OVER(ORDER BY date) as next_comm
FROM commission
WHERE show_id = 172;
```

Output :

show_id	date	commission	next_comm
172	1979-01-01	29.20	29.50
172	1979-01-01	29.50	8.25
172	1979-01-01	8.25	15.50
172	1979-01-01	15.50	10.25
172	1979-01-01	10.25	4.40
172	1979-01-01	4.40	80.20
172	1979-01-01	80.20	90.10
172	1979-01-02	90.10	25.50
172	1979-01-02	25.50	50.00
172	1979-01-02	50.00	20.20
172	1979-01-02	20.20	40.00
172	1979-01-02	40.00	null

12 rows selected (0.241 seconds)

Since there is no commission after 40.00, the last next_comm value is **null**.

#56daysofSQL

FIRST_VALUE()

The **FIRST_VALUE** window function returns the value of the specified expression with respect to the first row in the window frame.

General Syntax

```
FIRST_VALUE( expression )
OVER( [ PARTITION BY expr_list ]
      [ ORDER BY order_list ]
      [ frame_clause ] )
```

#56daysofSQL

Use Case Example of FIRST_VALUE()

The following example uses the **FIRST_VALUE** window function to identify the employee with the lowest sales for each dealer in Q1:

```
SELECT emp_name, dealer_id, sales,
FIRST_VALUE(sales) OVER(PARTITION BY dealer_id
ORDER BY sales) as dealer_low
FROM q1_sales;
```

Output :

emp_name	dealer_id	sales	dealer_low
Raphael Hull	1	8227	8227
Jack Salazar	1	9710	8227
Ferris Brown	1	19745	8227
Noel Meyer	1	19745	8227
Haviva Montoya	2	9308	9308
Beverly Lang	2	16233	9308
Kameko French	2	16233	9308
May Stout	3	9308	9308
Abel Kim	3	12369	9308
Ursa George	3	15427	9308

10 rows selected (0.299 seconds)

dealer_low shows the first sales value for each partitioned by dealer_id.

#56daysofSQL

LAST_VALUE()

The **LAST_VALUE** window function returns the value of the specified expression with respect to the last row in the window frame.

General Syntax

```
LAST_VALUE( expression )
OVER( [ PARTITION BY expr_list ]
      [ ORDER BY order_list ]
      [ frame_clause ] )
```

#56daysofSQL

Use Case Example of FIRST_VALUE()

The following example uses the **LAST_VALUE** window function to identify the last car sale each employee made at each dealership in 2013:

```
SELECT emp_name, dealer_id, sales, year,
LAST_VALUE(sales) OVER(PARTITION BY  emp_name
ORDER BY year) as last_sale
FROM emp_sales
WHERE year = 2013;
```

Output :

emp_name	dealer_id	sales	year	last_sale
Beverly Lang	2	5324	2013	5324
Ferris Brown	1	22003	2013	22003
Haviva Montoya	2	6345	2013	13100
Haviva Montoya	2	13100	2013	13100
Kameko French	2	7540	2013	7540
May Stout	2	4924	2013	15000
May Stout	2	8000	2013	15000
May Stout	2	15000	2013	15000
Noel Meyer	1	13314	2013	13314
Raphael Hull	1	-4000	2013	14000
Raphael Hull	1	14000	2013	14000
Ursa George	1	10865	2013	10865

12 rows selected (0.284 seconds)

last_sale shows the last sales value made by each employee.