Window-Funktionen

Prof. Dr. Ingo Claßen

Dienstag, 8. Juni 2021 10:32

- Aggregationen
- Rangbildungen
- Gleitende Durchschnitte
- Positionierungen
- Verteilungen

Daten

Prof. Dr. Ingo Claßen

Dienstag, 8. Juni 2021 10:32

Tabelle wwww

yyyy	mm	v1	v2	v3	v4	v5
2019	01	100	100	100	a	a
2019	02	200	100	100	b	a
2019	03	300	100	100	a	a
2019	04	400	100	400	b	a
2019	05	400	100	400	a	a
2019	06	600	100	400	a	a
2019	07	700	200	400	a	a
2019	08	800	200	300	a	a
2019	09	900	200	300	a	a
2019	10	100	200	300	a	a
2019	11	200	200	300	a	a
2019	12	300	200	300	a	a
2020	01	400	300	300	a	a
2020	02	500	300	300	b	a
2020	03	600	300	400	a	a
2020	04	600	300	400	a	a
2020	05	800	300	400	a	a
2020	06	900	300	400	b	a
2020	07	100	400	400	b	a
2020	08	200	400	400	a	a
2020	09	300	400	400	a	a
2020	10	400	400	400	a	a
2020	11	500	400	400	a	a
2020	12	600	400	400	a	a
2021	01	700	500	400	a	a
2021	02	800	500	400	a	a
2021	03	900	500	200	a	a
2021	04	100	500	200	b	a
2021	05	200	500	200	a	a
2021	06	300	500	200	a	a
2021	07	400	600	100	a	a
2021	08	500	600	100	b	a
2021	09	600	600	100	a	a
2021	10	700	600	100	a	a
2021	11	800	500	100	b	a
2021	12	900	500	100	a	a

Aggregationen 1

Prof. Dr. Ingo Claßen

Mittwoch, 3. August 2022 09:17

Fenster:

partition by	gesamte Partition
order by	von Beginn bis zur aktuellen Zeile (bis zum letzten Peer)
partition by order by	von Beginn der Partition bis zur aktuellen Zeile in der Partition (bis zum letzten Peer)

```
select
  yyyy,
  mm,
  v1,
  sum(v1) over (partition by yyyy order by mm) as kumuliert,
  sum(v1) over (partition by yyyy) as jahr,
  sum(v1) over (order by yyyy, mm) as alles_kumuliert,
  sum(v1) over () as alles
from wwww
where yyyy in ('2020', '2021') and mm in ('01', '02', '03', '04')
order by yyyy, mm
;
```

yyyy	mm	v1	kumuliert	jahr	alles_kumuliert	alles
2020	01	400	400	2.100	400	4.600
2020	02	500	900	2.100	900	4.600
2020	03	600	1.500	2.100	1.500	4.600
2020	04	600	2.100	2.100	2.100	4.600
2021	01	700	700	2.500	2.800	4.600
2021	02	800	1.500	2.500	3.600	4.600
2021	03	900	2.400	2.500	4.500	4.600
2021	04	100	2.500	2.500	4.600	4.600

Window-Funktionen in Berechnungen, hier prozentualer Anteil

```
select
  yyyy,
  mm,
  v1,
  sum(v1) over (partition by yyyy) as jahr,
  round((cast(v1 as numeric) / sum(v1) over (partition by yyyy)) * 100, 2) as prozent
from wwww
where yyyy in ('2021')
order by yyyy, mm
;
```

yyyy	mm	v1	jahr	prozent
2021	01	700	6.900	10,14
2021	02	800	6.900	11,59
2021	03	900	6.900	13,04
2021	04	100	6.900	1,45
2021	05	200	6.900	2,9
2021	06	300	6.900	4,35
2021	07	400	6.900	5,8
2021	08	500	6.900	7,25
2021	09	600	6.900	8,7
2021	10	700	6.900	10,14
2021	11	800	6.900	11,59
2021	12	900	6.900	13,04

Sortierung über v2

```
select
  yyyy,
  mm,
  v2,
  sum(v2) over (order by v2) as vv2
from wwww
where yyyy in ('2021')
order by v2
;
```

yyyy	mm	v2	vv2
2021	01	500	4.000
2021	02	500	4.000
2021	03	500	4.000
2021	04	500	4.000
2021	05	500	4.000
2021	06	500	4.000
2021	11	500	4.000
2021	12	500	4.000
2021	09	600	6.400
2021	10	600	6.400
2021	07	600	6.400
2021	08	600	6.400

```
select
  yyyy,
  mm,
  v1,
  rank() over (order by v1 desc) as rang,
  dense_rank() over (order by v1 desc) as dichter_rang,
  percent_rank() over (order by v1 desc) as prozent_rang,
  row_number() over (order by v1 desc) as zeilennummer
from wwww
where yyyy in ('2021')
;
```

yyyy	mm	v1	rang	dichter_rang	prozent_rang	zeilennummer
2021	03	900	1	1	0	1
2021	12	900	1	1	0	2
2021	11	800	3	2	0,1818181818	3
2021	02	800	3	2	0,1818181818	4
2021	01	700	5	3	0,3636363636	5
2021	10	700	5	3	0,3636363636	6
2021	09	600	7	4	0,5454545455	7
2021	08	500	8	5	0,6363636364	8
2021	07	400	9	6	0,7272727273	9
2021	06	300	10	7	0,8181818182	10
2021	05	200	11	8	0,9090909091	11
2021	04	100	12	9	1	12

```
select
  yyyy,
  mm,
  v1,
  rank() over (order by v1 desc) as rang,
  dense_rank() over (order by v1 desc) as dichter_rang,
  percent_rank() over (order by v1 desc) as prozent_rang,
  row_number() over (order by v1 desc) as zeilennummer
from wwww
where yyyy in ('2021')
order by yyyy, mm
;
```

Andere Sortierung in der Ausgabe als in der Rangbildung

yyyy	mm	v1	rang	dichter_rang	prozent_rang	zeilennummer
2021	01	700	5	3	0,3636363636	5
2021	02	800	3	2	0,1818181818	4
2021	03	900	1	1	0	1
2021	04	100	12	9	1	12
2021	05	200	11	8	0,9090909091	11
2021	06	300	10	7	0,8181818182	10
2021	07	400	9	6	0,7272727273	9
2021	08	500	8	5	0,6363636364	8
2021	09	600	7	4	0,5454545455	7
2021	10	700	5	3	0,3636363636	6
2021	11	800	3	2	0,1818181818	3
2021	12	900	1	1	0	2

Rangbildung pro Partition

```
select
  yyyy,
  mm,
  v1,
  rank() over (partition by yyyy order by v1 desc) as rang,
  dense_rank() over (partition by yyyy order by v1 desc) as dichter_rang,
  row_number() over (partition by yyyy order by v1 desc) as zeilennummer
from wwww
where yyyy in ('2020', '2021')
;
```

yyyy	mm	v1	rang	dichter_rang	zeilennummer
2020	06	900	1	1	1
2020	05	800	2	2	2
2020	04	600	3	3	3
2020	03	600	3	3	4
2020	12	600	3	3	5
2020	02	500	6	4	6
2020	11	500	6	4	7
2020	01	400	8	5	8
2020	10	400	8	5	9
2020	09	300	10	6	10
2020	08	200	11	7	11
2020	07	100	12	8	12
2021	12	900	1	1	1
2021	03	900	1	1	2
2021	11	800	3	2	3
2021	02	800	3	2	4
2021	01	700	5	3	5
2021	10	700	5	3	6
2021	09	600	7	4	7
2021	08	500	8	5	8
2021	07	400	9	6	9
2021	06	300	10	7	10
2021	05	200	11	8	11
2021	04	100	12	9	12

```
select
  yyyy,
  mm,
  v1,
  avg(v1) over (order by mm rows between 1 preceding and 1 following)
as jahr
from wwww
where yyyy in ('2019')
order by yyyy, mm
;
```

yyyy	mm	v1	jahr
2019	01	100	150
2019	02	200	200
2019	03	300	300
2019	04	400	366,6666666667
2019	05	400	466,6666666667
2019	06	600	566,6666666667
2019	07	700	700
2019	08	800	800
2019	09	900	600
2019	10	100	400
2019	11	200	200
2019	12	300	250

Einschränkung des Fensters:

- UNBOUNDED PRECEDING
- *offset* PRECEDING
- CURRENT ROW
- *offset* FOLLOWING
- UNBOUNDED FOLLOWING

```
select
  mm,
  v1,
  lag(mm) over (order by mm) as eins_vorher,
  lag(mm, 4) over (order by mm) as vier_vorher,
  first_value(mm) over (order by mm) as erster,
  last_value(mm) over (order by mm rows between unbounded preceding and unbounded following) as letzter,
  nth_value(mm, 3) over (order by mm rows between unbounded preceding and unbounded following) as dritter
from wwww
where yyyy in ('2021')
order by mm
;
```

mm	v1	eins_vorher	vier_vorher	erster	letzter	dritter
01	700	[NULL]	[NULL]	01	12	03
02	800	01	[NULL]	01	12	03
03	900	02	[NULL]	01	12	03
04	100	03	[NULL]	01	12	03
05	200	04	01	01	12	03
06	300	05	02	01	12	03
07	400	06	03	01	12	03
08	500	07	04	01	12	03
09	600	08	05	01	12	03
10	700	09	06	01	12	03
11	800	10	07	01	12	03
12	900	11	08	01	12	03

```
select
  mm,
  v1,
  lag(v1) over (order by mm) as eins_vorher,
  lag(v1, 4) over (order by mm) as vier_vorher,
  first_value(v1) over (order by mm) as erster,
  last_value(v1) over (order by mm rows between unbounded preceding and unbounded following) as letzter,
  nth_value(v1, 3) over (order by mm rows between unbounded preceding and unbounded following) as dritter
from wwww
where yyyy in ('2021')
order by mm
;
```

Spalte v1 in der Ausgabe

mm	v1	eins_vorher	vier_vorher	erster	letzter	dritter
01	700	[NULL]	[NULL]	700	900	900
02	800	700	[NULL]	700	900	900
03	900	800	[NULL]	700	900	900
04	100	900	[NULL]	700	900	900
05	200	100	700	700	900	900
06	300	200	800	700	900	900
07	400	300	900	700	900	900
08	500	400	100	700	900	900
09	600	500	200	700	900	900
10	700	600	300	700	900	900
11	800	700	400	700	900	900
12	900	800	500	700	900	900

Year over Year

```
select
  yyyy,
  mm,
  lag(yyyy || '-' || mm) over (partition by mm order by yyyy, mm) as ein_jahr_vorher,
  lag(yyyy || '-' || mm, 2) over (partition by mm order by yyyy, mm) as zwei_jahre_vorher
from wwww
where mm in ('01', '02', '08')
order by yyyy, mm
;
```

yyyy	mm	ein_jahr_vorher	zwei_jahre_vorher
2019	01	[NULL]	[NULL]
2019	02	[NULL]	[NULL]
2019	08	[NULL]	[NULL]
2020	01	2019-01	[NULL]
2020	02	2019-02	[NULL]
2020	08	2019-08	[NULL]
2021	01	2020-01	2019-01
2021	02	2020-02	2019-02
2021	08	2020-08	2019-08

```
select
  yyyy,
  mm,
  v3,
  row_number() over (order by v3) as nr,
  ntile(6) over (order by v3) as bucket,
  cume_dist() over (order by v3) as verteilung
from wwww
where yyyy in ('2020', '2021')
order by v3
;
```

yyyy	mm	v3	nr	bucket	verteilung
2021	08	100	1	1	0,25
2021	07	100	2	1	0,25
2021	12	100	3	1	0,25
2021	11	100	4	1	0,25
2021	10	100	5	2	0,25
2021	09	100	6	2	0,25
2021	05	200	7	2	0,4166666667
2021	03	200	8	2	0,4166666667
2021	06	200	9	3	0,4166666667
2021	04	200	10	3	0,4166666667
2020	01	300	11	3	0,5
2020	02	300	12	3	0,5
2021	01	400	13	4	1
2021	02	400	14	4	1
2020	12	400	15	4	1
2020	11	400	16	4	1
2020	10	400	17	5	1
2020	09	400	18	5	1
2020	08	400	19	5	1
2020	07	400	20	5	1
2020	06	400	21	6	1
2020	05	400	22	6	1
2020	04	400	23	6	1
2020	03	400	24	6	1

Verteilungen pro Partition

```
select
  yyyy,
  mm,
  v3,
  row_number() over (partition by yyyy order by v3) as nr,
  ntile(6) over (partition by yyyy order by v3) bucket,
  cume_dist() over (partition by yyyy order by v3) as verteilung
from wwww
where yyyy in ('2020', '2021')
order by yyyy, v3
;
```

yyyy	mm	v3	nr	bucket	verteilung
2020	01	300	1	1	0,1666666667
2020	02	300	2	1	0,1666666667
2020	03	400	3	2	1
2020	04	400	4	2	1
2020	05	400	5	3	1
2020	06	400	6	3	1
2020	07	400	7	4	1
2020	08	400	8	4	1
2020	09	400	9	5	1
2020	10	400	10	5	1
2020	11	400	11	6	1
2020	12	400	12	6	1
2021	12	100	1	1	0,5
2021	07	100	2	1	0,5
2021	08	100	3	2	0,5
2021	09	100	4	2	0,5
2021	10	100	5	3	0,5
2021	11	100	6	3	0,5
2021	03	200	7	4	0,8333333333
2021	04	200	8	4	0,8333333333
2021	05	200	9	5	0,8333333333
2021	06	200	10	5	0,8333333333
2021	01	400	11	6	1
2021	02	400	12	6	1