

Агрегиращи функции

1. изискват групиране (в повечето случаи) на данните за да работят правилно

При групиране на данни, сървърът сравнява данните в съответните колони по редове и ако открие два или повече реда със съвпадащи стойности, представя редовете като група, оставяйки в резултат само един от съвпадащите редове.

Как да групираме данните? е най-важният въпрос, на който трябва да отговорим, преди да използваме която и да е от агрегиращите функции.

Пример 1

В колона [3] всички стойности по редовете са уникални т.е. групиране не може да се получи => използването, на която и да е от агрегатните функции няма да върне полезен резултат.

<1>	<2>	<3>
a	b	e
a	b	f
a	c	g
a	c	i
a	d	j

```
SELECT <1>, <2>, <3>, COUNT(*) N
FROM ...
GROUP BY <1>, <2>, <3>
```

<1>	<2>	<3>	N
a	b	e	1
a	b	f	1
a	c	g	1
a	c	i	1
a	d	j	1

Пример 2

Първите два реда и следващите два реда се повтарят, а последният ред е единствен в групата => имаме две групи с по два реда и една с един ред.

<1>	<2>
a	b
a	b

a	c
a	c
a	d

```
SELECT <1>, <2>, COUNT(*) N
FROM ...
GROUP BY <1>, <2>
```

<1>	<2>	N
a	b	2
a	c	2
a	d	1

Пример 3

Всички редове съдържат една и съща стойност => имаме 1 група от 5 реда

<1>
a
a
a
a
a

```
SELECT <1>, COUNT(*) N
FROM ...
GROUP BY <1>
```

<1>	N
a	5

Основни агрегиращи функции

Функция	Действие
COUNT()	брои редове
AVG()	изчислява средно аритметично
SUM()	сумира

MIN()	минимум	
MAX()	максимум	

```
SELECT
  customer_id
,   COUNT(*) n_orders
FROM
  orders
--WHERE
--   customer_id = 1 -- ALFKI
GROUP BY
  customer_id
ORDER BY
  2 DESC
```

Филтриране на резултати от агрегираща функция

```
SELECT <1>, <2>, COUNT(*)
FROM ...
GROUP BY <1>, <2>
HAVING COUNT(*) < 5
```

HAVING трябва да се използва САМО за филтриране на резултати от агрегираща функция. За обикновените колони и за колоните съдържащи други функции е по-добре да се използва клаузата **WHERE**.

```
SELECT
  customer_id
,   COUNT(*) n_orders
FROM
  orders
--WHERE
--   customer_id = 1 -- ALFKI
GROUP BY
  customer_id
HAVING
  COUNT(*) < 3
ORDER BY
  2 DESC

-----
-
SELECT
  t1.customer_code
  -- грешен формат на COUNT()
,   COUNT(*) n_orders
FROM
  customers t1
  LEFT OUTER JOIN
```

```

    orders t2
      ON t2.customer_id = t1.customer_id
--WHERE
--    customer_id = 1 -- ALFKI
GROUP BY
    t1.customer_code
HAVING
    COUNT(*) < 3
ORDER BY
    2 DESC

```

- COUNT(*)

<1>

A
B
NULL
NULL
C

```

SELECT COUNT(*) N
FROM ...

```

N

5

- COUNT(име_на_колона)

<1>

A
B
NULL
NULL
C

```

SELECT COUNT(<1>) N
FROM ...

```

```
| N |
|---|
| 3 |
```

```
SELECT
    t1.customer_code
,   COUNT(t2.order_id) n_orders
FROM
    customers t1
    LEFT OUTER JOIN
    orders t2
    ON t2.customer_id = t1.customer_id
--WHERE
--    customer_id = 1 -- ALFKI
GROUP BY
    t1.customer_code
HAVING
    COUNT(t2.order_id) < 3
ORDER BY
    2 DESC
-----
-
SELECT
    t1.customer_id
,   t1.order_id
,   t1.order_date
,   SUM(t2.unit_price * t2.quantity) total
,   COUNT(*) n_items
FROM
    orders t1
    INNER JOIN
    order_details t2
    ON t2.order_id = t1.order_id
WHERE
    t1.customer_id = 1 -- ALFKI
GROUP BY
    t1.customer_id
,   t1.order_id
,   t1.order_date
ORDER BY
    2
-----
-
SELECT
    t1.customer_id
,   SUM(t2.unit_price * t2.quantity) total
,   COUNT(*) n_items
FROM
    orders t1
    INNER JOIN
    order_details t2
    ON t2.order_id = t1.order_id
```

```

WHERE
    t1.customer_id = 1 -- ALFKI
GROUP BY
    t1.customer_id
ORDER BY
    2
-----
-
SELECT
    vt.customer_id
,   MIN(vt.total) min_total
,   ROUND(AVG(vt.total),2) avg_total
,   MAX(vt.total) max_total
,   MIN(vt.n_items) min_items
,   ROUND(AVG(vt.n_items)) AVG_items
,   MAX(vt.n_items) max_items
FROM
    (SELECT
        t1.customer_id
      ,   t1.order_id
      ,   SUM(t2.unit_price * t2.quantity) total
      ,   COUNT(*) n_items
    FROM
        orders t1
        INNER JOIN
        order_details t2
        ON t2.order_id = t1.order_id
    -- WHERE
    --     t1.customer_id = 1 -- ALFKI
    GROUP BY
        t1.customer_id
      ,   t1.order_id
    ) vt
GROUP BY
    vt.customer_id
ORDER BY
    2
-----
-
SELECT
    vt.customer_id
,   MIN(vt.total) min_total
,   ROUND(AVG(vt.total),2) avg_total
,   MAX(vt.total) max_total
,   MIN(vt.n_items) min_items
,   ROUND(AVG(vt.n_items)) AVG_items
,   MAX(vt.n_items) max_items
-- допълнително --
,   SUM(vt.total) grand_total
,   SUM(vt.n_items) total_items
,   COUNT(*) n_orders
FROM
    (SELECT
        t1.customer_id
      ,   t1.order_id

```

```
,    SUM(t2.unit_price * t2.quantity) total
,    COUNT(*) n_items
FROM
    orders t1
    INNER JOIN
    order_details t2
    ON t2.order_id = t1.order_id
-- WHERE
--     t1.customer_id = 1 -- ALFKI
GROUP BY
    t1.customer_id
,    t1.order_id
) vt
GROUP BY
    vt.customer_id
ORDER BY
    grand_total DESC
```