



# Ранжирующие функции

# Ранжирующие функции

Функции	Описание
row_number	Возвращает номер строки
rank	Возвращает ранг каждой строки. В данном случае значения уже анализируются и, в случае нахождения одинаковых, функция возвращает одинаковый ранг с пропуском следующего значения
dense_rank	Возвращает ранг каждой строки. Но в отличие от функции RANK, для одинаковых значений она возвращает ранг, не пропуская следующий
ntile	Позволяет определить, к какой группе относится текущая строка. Количество групп задаётся в скобках

# Функции row\_number, rank(), dense\_rank()

```
select cnt,  
row_number() over (order by cnt) cnt_row_number,  
rank() over (order by cnt) cnt_rank,  
dense_rank() over (order by cnt) cnt__dense_rank  
from skill_orders order by cnt
```

cnt	cnt_row_number	cnt_rank	cnt__dense_rank
1	1	1	1
1	2	1	1
2	3	3	2
3	4	4	3
4	5	5	4
4	6	5	4
5	7	7	5
5	8	7	5
5	9	7	5
8	10	10	6
9	11	11	7
11	12	12	8
12	13	13	9
13	14	14	10
14	15	15	11
15	16	16	12
17	17	17	13
18	18	18	14
18	19	18	14
19	20	20	15
20	21	21	16
21	22	22	17
23	23	23	18
25	24	24	19
26	25	25	20
26	26	25	20
26	27	25	20

# Функции rank(), dense\_rank()

```
select cnt,  
row_number() over (order by cnt) cnt_row_number,  
rank() over (order by cnt) cnt_rank,  
dense_rank() over (order by cnt) cnt__dense_rank  
from skill_orders order by cnt
```

cnt	cnt_row_number	cnt_rank	cnt__dense_rank
1	1	1	1
1	2	1	1
2	3	3	2
3	4	4	3
4	5	5	4
4	6	5	4
5	7	7	5
5	8	7	5
5	9	7	5
8	10	10	6
9	11	11	7
11	12	12	8
12	13	13	9
13	14	14	10
14	15	15	11
15	16	16	12
17	17	17	13
18	18	18	14
18	19	18	14
19	20	20	15
20	21	21	16
21	22	22	17
23	23	23	18
25	24	24	19
26	25	25	20
26	26	25	20
26	27	25	20

# Функции rank(), dense\_rank()

```
select cnt,  
row_number() over (order by cnt) cnt_row_number,  
rank() over (order by cnt) cnt_rank,  
dense_rank() over (order by cnt) cnt__dense_rank  
from skill_orders order by cnt
```

cnt	cnt_row_number	cnt_rank	cnt__dense_rank
1	1	1	1
1	2	1	1
2	3	→ 3	2
3	4	4	3
4	5	5	4
4	6	5	4
5	7	→ 7	5
5	8	7	5
5	9	7	5
8	10	→ 10	6
9	11	11	7
11	12	12	8
12	13	13	9
13	14	14	10
14	15	15	11
15	16	16	12
17	17	17	13
18	18	18	14
18	19	18	14
19	20	20	15
20	21	21	16
21	22	22	17
23	23	23	18
25	24	24	19
26	25	→ 25	20
26	26	25	20
26	27	25	20

# Функции rank(), dense\_rank()

```
select cnt,  
row_number() over (order by cnt) cnt_row_number,  
rank() over (order by cnt) cnt_rank,  
dense_rank() over (order by cnt) cnt_dense_rank  
from skill_orders order by cnt
```

cnt	cnt_row_number	cnt_rank	cnt_dense_rank
1	1	1	1
1	2	1	1
2	3	3	2
3	4	4	3
4	5	5	4
4	6	5	4
5	7	7	5
5	8	7	5
5	9	7	5
8	10	10	6
9	11	11	7
11	12	12	8
12	13	13	9
13	14	14	10
14	15	15	11
15	16	16	12
17	17	17	13
18	18	18	14
18	19	18	14
19	20	20	15
20	21	21	16
21	22	22	17
23	23	23	18
25	24	24	19
26	25	25	20
26	26	25	20
26	27	25	20

+1

+1

+1

# Функции ntile()

Функция	Описание
ntile(n)	Делит результирующий набор на группы по определённому столбцу. Количество групп указывается в качестве параметра

```
select cnt, ntile(4) over (order by cnt) from skill_orders order by cnt
```



# Практика к уроку

Напишите SQL-запрос, возвращающий все строки и все колонки таблицы с продажами **skill\_sales**

К набору колонок добавьте две вычисляемые колонки:

Название колонки	Содержимое
<b>total_rank</b>	Колонка должна содержать ранги, без пропусков. Первый ранг должен быть у продажи на самую большую сумму и далее по уменьшению суммы сделки
<b>document_type_rank</b>	Колонка должна содержать ранги, без пропусков. Ранги присваиваются для каждой группы продаж по типу документа отдельно. Первый ранг должен быть у сделки на самую большую сумму в группе и далее по уменьшению суммы сделки