Аналітичні функції (продовження)

Лекція №3

Rank()

```
2  select
3    department_id
4    ,last_name
5    ,salary
6    ,rank() over(partition by department_id
7                 order by salary desc) as salary_rank
8  from employees
9  order by department_id, salary desc;
```

Query Result

SQL | Fetched 50 rows in 0,455 seconds

	DEPARTMENT_ID	LAST_NAME	SALARY	SALARY_RANK
16	50	Sarchand	4200	6
17	50	Bull	4100	7
18	50	Bell	4000	8
19	50	Everett	3900	9
20	50	Chung	3800	10
21	50	Dilly	3600	11
22	50	Ladwig	3600	11
23	50	Rajs	3500	13

Dense_rank()

```
11  -- Example 3.2
12  select
13    department_id
14    ,last_name
15    ,salary
16    ,dense_rank() over(partition by department_id
17                       order by salary desc) as salary_rank
18  from employees
19  order by department_id, salary desc;
```

Query Result

SQL | Fetched 50 rows in 0,208 seconds

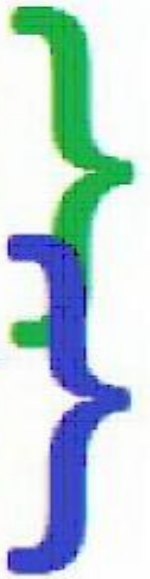
	DEPARTMENT_ID	LAST_NAME	SALARY	SALARY_RANK
16	50	Sarchand	4200	6
17	50	Bull	4100	7
18	50	Bell	4000	8
19	50	Everett	3900	9
20	50	Chung	3800	10
21	50	Dilly	3600	11
22	50	Ladwig	3600	11
23	50	Rais	3500	12

Sliding window

When data set is ordered, we can ask questions about:

- all rows "before" or "after" current;
- several previous rows;
- rows "around" current (several before and several after);
- etc.

LAST_NAME	HIRE_DATE	SALARY
Hunold	08.01.2012	300.00
Ernst	15.01.2012	600.00
Austin	22.01.2012	600.00
Pataballa	29.01.2012	900.00
Lorentz	05.02.2012	900.00
Greenberg	12.02.2012	1200.00
Faviet	19.02.2012	1200.00
Chen	26.02.2012	1500.00
Sciarra	04.03.2012	1500.00
Urman	11.03.2012	1800.00
Popp	18.03.2012	1800.00
Doe	25.03.2012	2100.00



Sliding window

Beyond default, sliding windows can be specified in:

- rows (several rows before and several after);
- range (some range before/after/around current row, based on ordering criteria).

Rows between n preceding and m following

```
-- Example 3.3
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,avg(salary) over(partition by department_id
                    order by hire_date
                    rows between 1 preceding and 1 following )
   as sliding_avg
from emp
order by department_id;
```

Result of 3.3

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG
1	10	Whalen	17.09.03	4400	4400
2	20	Hartstein	17.02.04	13000	9500
3	20	Fay	17.08.05	6000	9500
4	30	Raphaely	07.12.02	11000	7050
5	30	Khoo	18.05.03	3100	5633,33333333333333333333333333333333333
6	30	Tobias	24.07.05	2800	2933,33333333333333333333333333333333333
7	30	Baida	24.12.05	2900	2766,66666666666666666666666666666666667
8	30	Himuro	15.11.06	2600	2666,66666666666666666666666666666666667
9	30	Colmenares	10.08.07	2500	2550
10	40	Mavris	07.06.02	6500	6500
11	50	Kaufling	01.05.03	7900	5750
12	50	Ladwig	14.07.03	3600	5000
13	50	Rajs	17.10.03	3500	3766,66666666666666666666666666666666667
14	50	Sarchand	27.01.04	4200	3900
15	50	Bell	04.02.04	4000	3833,33333333333333333333333333333333333
16	50	Mallin	14.06.04	3300	5100
17	50	Weiss	18.07.04	8000	4800

Питання:

1) Як змінити запит, щоб після коми у середньому було лише 2 знаки?
2) Чим відрізняється order by в over() від order by в кінці запита?
3) Чи можна використовувати аліас для поля, що було обчислене за допомогою аналітичних функцій в умові where?
4) Чим відрізняються

   max(salary) over()

від

   max(salary) over(partition by department_id) ?

Range between n preceding and m following

```
-- Example 3.4
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,avg(salary) over(partition by department_id
                    order by hire_date
                    range between 15 preceding and 15 following ) as
     sliding_avg
from emp
order by department_id;
```

Result of 3.4

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG
4	30	Raphaely	07.12.02	11000	11000
5	30	Khoo	18.05.03	3100	3100
6	30	Tobias	24.07.05	2800	2800
7	30	Baida	24.12.05	2900	2900
8	30	Himuro	15.11.06	2600	2600
9	30	Colmenares	10.08.07	2500	2500
10	40	Mavris	07.06.02	6500	6500
11	50	Kaufling	01.05.03	7900	7900
12	50	Ladwig	14.07.03	3600	3600
13	50	Rajs	17.10.03	3500	3500
14	50	Sarchand	27.01.04	4200	4100
15	50	Bell	04.02.04	4000	4100
16	50	Mallin	14.06.04	3300	3300
17	50	Weiss	18.07.04	8000	8000
18	50	Davies	29.01.05	3100	3100
19	50	Marlow	16.02.05	2500	3500
20	50	Bull	20.02.05	4100	3500

Salary range

```
-- Example 3.5
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,avg(salary) over(partition by department_id
                    order by salary
                    range between 1000 preceding and 1000 following
     ) as sliding_avg
from employees
order by department_id;
```

Result of 3.5

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG
1	10	Whalen	17.09.03	4400	4400
2	20	Fay	17.08.05	6000	6000
3	20	Hartstein	17.02.04	13000	13000
4	30	Colmenares	10.08.07	2500	2780
5	30	Himuro	15.11.06	2600	2780
6	30	Tobias	24.07.05	2800	2780
7	30	Baida	24.12.05	2900	2780
8	30	Khoo	18.05.03	3100	2780
9	30	Raphaely	07.12.02	11000	11000
10	40	Mavris	07.06.02	6500	6500
11	50	Olson	10.04.07	2100	2660
12	50	Philtanker	06.02.08	2200	2734,48275862068965517241379310344827586
13	50	Markle	08.03.08	2200	2734,48275862068965517241379310344827586
14	50	Gee	12.12.07	2400	2790,625
15	50	Landry	14.01.07	2400	2790,625
16	50	Patel	06.04.06	2500	2812,12121212121212121212121212121212121212
17	50	Vargas	09.07.06	2500	2812,12121212121212121212121212121212121212

Питання:

1) Як би виглядав попередній запит, якщо замість 1000 поставити 15?
2) Як написати запит, щоб побачити кількість людей з однаковою зарплатою?

Відповідь на питання 2)

```
-- Example 3.6
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,avg(salary) over(partition by department_id
                    order by salary
                    range between 15 preceding and 15 following ) as sliding_avg
  ,count(*) over(order by salary
                 range between 15 preceding and 15 following ) as sliding_count
from employees
order by department_id;
```

Result of 3.6

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG	SLIDING_COUNT
1	10	Whalen	17.09.03	4400	4400	1
2	20	Fay	17.08.05	6000	6000	2
3	20	Hartstein	17.02.04	13000	13000	1
4	30	Colmenares	10.08.07	2500	2500	6
5	30	Himuro	15.11.06	2600	2600	4
6	30	Tobias	24.07.05	2800	2800	4
7	30	Baida	24.12.05	2900	2900	3
8	30	Khoo	18.05.03	3100	3100	4
9	30	Raphaely	07.12.02	11000	11000	3
10	40	Mavris	07.06.02	6500	6500	2
11	50	Olson	10.04.07	2100	2100	1
12	50	Philtanker	06.02.08	2200	2200	2
13	50	Markle	08.03.08	2200	2200	2
14	50	Landry	14.01.07	2400	2400	2
15	50	Gee	12.12.07	2400	2400	2
16	50	Sullivan	21.06.07	2500	2500	6
17	50	Vargas	09.07.06	2500	2500	6

Розширені можливості sliding window

```
-- Example 3.7
select
  department_id
 ,last_name
 ,hire_date
 ,salary
 ,avg(salary) over(partition by department_id
                   order by salary
                   range between 15 preceding and 15 following ) as sliding_avg
 ,count(*) over(order by salary
                range between 15 preceding and 15 following ) as sliding_count
 ,count(*) over(order by salary
                range current row ) as sliding_count_2
from employees
order by department_id;
```

Result of 3.7

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG	SLIDING_COUNT	SLIDING_COUNT_2
1	10	Whalen	17.09.03	4400	4400	1	1
2	20	Fay	17.08.05	6000	6000	2	2
3	20	Hartstein	17.02.04	13000	13000	1	1
4	30	Colmenares	10.08.07	2500	2500	6	6
5	30	Himuro	15.11.06	2600	2600	4	4
6	30	Tobias	24.07.05	2800	2800	4	4
7	30	Baida	24.12.05	2900	2900	3	3
8	30	Khoo	18.05.03	3100	3100	4	4
9	30	Raphaely	07.12.02	11000	11000	3	3
10	40	Mavris	07.06.02	6500	6500	2	2
11	50	Olson	10.04.07	2100	2100	1	1
12	50	Philtanker	06.02.08	2200	2200	2	2
13	50	Markle	08.03.08	2200	2200	2	2
14	50	Landry	14.01.07	2400	2400	2	2
15	50	Gee	12.12.07	2400	2400	2	2
16	50	Sullivan	21.06.07	2500	2500	6	6
17	50	Vargas	09.07.06	2500	2500	6	6

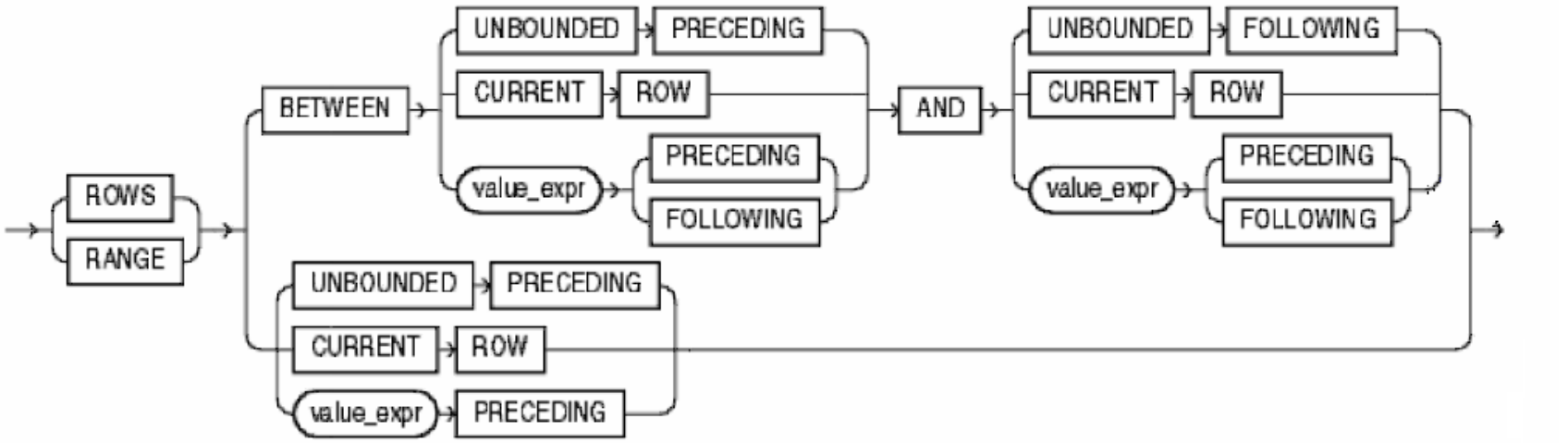
Ще один варіант

```
-- Example 3.8
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,avg(salary) over(partition by department_id
                    order by salary
                    range between 15 preceding and 15 following ) as sliding_avg
  ,count(*) over(order by salary
                 range between 15 preceding and 15 following ) as sliding_count
  ,count(*) over(order by salary
                 range current row ) as sliding_count_2
  ,count(*) over(partition by salary) as sliding_count_3
from employees
order by department_id;
```

Result of 3.8

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	SLIDING_AVG	SLIDING_COUNT	SLIDING_COUNT_2	SLIDING_COUNT_3
1	10	Whalen	17.09.03	4400	4400	1	1	1
2	20	Fay	17.08.05	6000	6000	2	2	2
3	20	Hartstein	17.02.04	13000	13000	1	1	1
4	30	Colmenares	10.08.07	2500	2500	6	6	6
5	30	Himuro	15.11.06	2600	2600	4	4	4
6	30	Tobias	24.07.05	2800	2800	4	4	4
7	30	Baida	24.12.05	2900	2900	3	3	3
8	30	Khoo	18.05.03	3100	3100	4	4	4
9	30	Raphaely	07.12.02	11000	11000	3	3	3
10	40	Mavris	07.06.02	6500	6500	2	2	2
11	50	Olson	10.04.07	2100	2100	1	1	1
12	50	Philtanker	06.02.08	2200	2200	2	2	2
13	50	Markle	08.03.08	2200	2200	2	2	2
14	50	Landry	14.01.07	2400	2400	2	2	2
15	50	Gee	12.12.07	2400	2400	2	2	2
16	50	Sullivan	21.06.07	2500	2500	6	6	6
17	50	Vargas	09.07.06	2500	2500	6	6	6

Complete windows syntax



Неявне вікно (default window)

```
-- Example 3.9
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,count(*) over(partition by department_id order
    by hire_date) as count_1
from employees
order by department_id;
```

Result of 3.9

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	COUNT_1
1	10	Whalen	17.09.03	4400	1
2	20	Hartstein	17.02.04	13000	1
3	20	Fay	17.08.05	6000	2
4	30	Raphaely	07.12.02	11000	1
5	30	Khoo	18.05.03	3100	2
6	30	Tobias	24.07.05	2800	3
7	30	Baida	24.12.05	2900	4
8	30	Himuro	15.11.06	2600	5
9	30	Colmenares	10.08.07	2500	6
10	40	Mavris	07.06.02	6500	1
11	50	Kaufling	01.05.03	7900	1
12	50	Ladwig	14.07.03	3600	2
13	50	Rajs	17.10.03	3500	3
14	50	Sarchand	27.01.04	4200	4
15	50	Bell	04.02.04	4000	5

Використання виразів

```
-- Example 3.10
select
   department_id
  ,last_name
  ,hire_date
  ,salary
  ,count(*) over(partition by department_id
                 order by salary
                 range between 0 preceding and 2*salary following)
    as count_1
from employees
order by department_id;
```

Result of 3.10

	DEPARTMENT_ID	LAST_NAME	HIRE_DATE	SALARY	COUNT_1
1	10	Whalen	17.09.03	4400	1
2	20	Fay	17.08.05	6000	2
3	20	Hartstein	17.02.04	13000	1
4	30	Colmenares	10.08.07	2500	5
5	30	Himuro	15.11.06	2600	4
6	30	Tobias	24.07.05	2800	3
7	30	Baida	24.12.05	2900	2
8	30	Khoo	18.05.03	3100	1
9	30	Raphaely	07.12.02	11000	1
10	40	Mavris	07.06.02	6500	1
11	50	Olson	10.04.07	2100	41
12	50	Philtanker	06.02.08	2200	41
13	50	Markle	08.03.08	2200	41
14	50	Gee	12.12.07	2400	39
15	50	Landry	14.01.07	2400	39
16	50	Patel	06.04.06	2500	37
17	50	Vargas	09.07.06	2500	37

Дякую за увагу!