

ANALYTIC FUNCTION I

	2001 10

1. Intro

2001

가

Tom Kyte

가

8.1.6

. Windowing

scott

SQL

ORACLE DBMS V 8.1.6

가 26

- a. (calculate a running total)
- b. (find percentages within a group)
- c. N (top -n queries)
- d. (compute a moving average)
- e. (perform ranking queries)

1)

A) ANALYTIC FUNCTION

```
select deptno as "   ", ename as "   ", sal as "   ",
       row_number() over (partition by deptno order by ename) as "   SEQ",
       sum(sal) over (partition by deptno order by ename) as "   SAL   ",
       row_number() over (order by deptno,ename) as "   SEQ",
       sum(sal) over (order by deptno,ename) as "   SAL   "
from emp
order by deptno, ename ;
```

		L	SEQ	SAL	SEQ	SAL
10	CLARK	2450	1	2450	1	2450
10	KING	5000	2	7450	2	7450
10	MILLER	1300	3	8750	3	8750
20	ADAMS	1100	1	1100	4	9850
20	FORD	3000	2	4100	5	12850
20	JONES	2975	3	7075	6	15825
20	SCOTT	3000	4	10075	7	18825
20	SMITH	800	5	10875	8	19625
30	ALLEN	1600	1	1600	9	21225
30	BLAKE	2850	2	4450	10	24075

30	JAMES	950	3	5400	11	25025
30	MARTIN	1250	4	6650	12	26275
30	TURNER	1500	5	8150	13	27775
30	WARD	1250	6	9400	14	29025

B)

```

Select detpno,ename,sal,
      (select count(ename) from emp e3
       where e3.deptno = e1.deptno and e3.ename <= emp.ename) as "      ",
      (select sum(sal) from emp e3
       where e3.deptno = e1.deptno and e3.ename <= e1.ename)
as "      ",
      (select sum(sal) from emp e2
       where e2.deptno < e1.deptno or (e2.deptno = e1.deptno and e2.ename <= e1.ename))
as "      "
from emp e1
order by deptno, ename;

```

2. Syntax & Definition

```

FUNCTION_NAME (<arg1>,<arg2>,<arg3>) OVER

```

```

-----
      analytic_clause
(<Partition - > <order by > <windowing >)
-----

```

```

) sum(sal) over
  (partition by deptno
   order by ename) as "      "

```

FUNCTION_NAME : analytic function

(<arg1>,<arg2>,<arg3>) : analytic function 0-3 arguments .

OVER : query set indicate . , from, where, group
by , having .

analytic_clause : select list order by analytic fuction .
query filter .

:

1. select order by .
2. execution plan WINDOW SORT .
3. ANAL FUC SQL ORDER BY . JOIN, WHERE, GROUP BY , HAVING .

4 ANALYTIC FUNCTION .

2_1. Function Clause

2_2. Partition Clause

PARTITION BY : 1 VALUE_EXPR query가 partition .
rows query

Value_expr : , , nonanalytic functions, function expressions .

2_3. Order By Clause

ORDER BY : Partition .
value_expr ordering sequence 가 .

) analytic function **ORDER BY** **ROWS** .
) expr . position alias .

ASC | DESC : ASC default .

NULLS FIRST | NULLS LAST : NULL row first ? Last ?
NULLS LAST ASC default ,
DESC NULLS FIRST가 default .

2_4. Windowing Clause

ROWS | RANGE : window(, row set)
row keyword . row .
Partition top bottom sliding .
ROWS : ROW WINDOW
RANGE : offset .
) Windowing order by .

ordering	가	.	order by	unique
ordering	.			

BETWEEN ... AND :

one end point	start point
end point CURRENT ROW	. (DEFAULT)

```
UNBOUNDED PRECEDING : Partition      row      ,      start
                        point      end point      .
```

```

UNBOUNDED FOLLOWING : Partition      row      ,      end
                        Point      start point      .

```

CURRENT ROW: **start point** , current row or value(ROW ? ,
RANGE ?) .
end point value expr PRECEDING .

```

end point      ,      current row or value(ROW ? ,
RANGE ? )      .
end point      value_expr FOLLOWING      .

```

Value_expr PRECEDING : Value_expr PRECEDING 가 start point end point
Value_expr FOLLOWING .

Value_expr FOLLOWING : Value_expr FOLLOWING 가 start point end point
Value_expr PRECEDING .

Conversion function .
(NUMTOYMINTERVAL, NUMTODSINTERVAL)

ROWS

1. Value_expr 가 OFFSET . .
2. Value_expr 가 start point end point row 가 .

RANGE

1. Value_expr OFFSET . interval literal 가 constant
expression .
 2. ORDER BY 1 expression .
 3. Value_expr 가 ORDER BY expr NUMBER OR DATE datatype .

4. Value_expr interval value ORDER BY expr DATE datatype .

WINDOWING_CLAUSE 가 ,

RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

2_4_1. Range Windows

1. Asc

```
select  ename as "      ", sal as "      ", hiredate -100 as "      100      ",
        hiredate as "      ",
        first_value(ename) over
          (order by hiredate asc range 100 preceding) as "      _pre",
        first_value(hiredate) over
          (order by hiredate asc range 100 preceding) as "      _pre"
from emp
order by hiredate asc;
```

ENAME	SAL	100		_pre	_pre
SMITH	800	9/8/1980	12/17/1980	SMITH	12/17/1980
ALLEN	1600	11/12/1980	2/20/1981	SMITH	12/17/1980
WARD	1250	11/14/1980	2/22/1981	SMITH	12/17/1980
JONES	2975	12/23/1980	4/2/1981	ALLEN	2/20/1981
BLAKE	2850	1/21/1981	5/1/1981	ALLEN	2/20/1981
CLARK	2450	3/1/1981	6/9/1981	JONES	4/2/1981
TURNER	1500	5/31/1981	9/8/1981	CLARK	6/9/1981
MARTIN	1250	6/20/1981	9/28/1981	TURNER	9/8/1981
KING	5000	8/9/1981	11/17/1981	TURNER	9/8/1981
FORD	3000	8/25/1981	12/3/1981	TURNER	9/8/1981
JAMES	950	8/25/1981	12/3/1981	TURNER	9/8/1981
MILLER	1300	10/15/1981	1/23/1982	KING	11/17/1981
SCOTT	3000	1/9/1987	4/19/1987	SCOTT	4/19/1987
ADAMS	1100	2/12/1987	5/23/1987	SCOTT	4/19/1987

2. Desc

```
select  ename as "      ", sal as "      ",hiredate as "      ",
        hiredate+100 as "      100      ",
        first_value(ename) over
          (order by hiredate desc range 100 preceding) as "      _pre",
        first_value(hiredate) over
          (order by hiredate desc range 100 preceding) as "      _pre"
```

from emp
order by hiredate desc;

			100	_pre	_pre
ADAMS	1100	5/23/1987	8/31/1987	ADAMS	5/23/1987
SCOTT	3000	4/19/1987	7/28/1987	ADAMS	5/23/1987
MILLER	1300	1/23/1982	5/3/1982	MILLER	1/23/1982
FORD	3000	12/3/1981	3/13/1982	MILLER	1/23/1982
JAMES	950	12/3/1981	3/13/1982	MILLER	1/23/1982
KING	5000	11/17/1981	2/25/1982	MILLER	1/23/1982
MARTIN	1250	9/28/1981	1/6/1982	FORD	12/3/1981
TURNER	1500	9/8/1981	12/17/1981	FORD	12/3/1981
CLARK	2450	6/9/1981	9/17/1981	TURNER	9/8/1981
BLAKE	2850	5/1/1981	8/9/1981	CLARK	6/9/1981
JONES	2975	4/2/1981	7/11/1981	CLARK	6/9/1981
WARD	1250	2/22/1981	6/2/1981	BLAKE	5/1/1981
ALLEN	1600	2/20/1981	5/31/1981	BLAKE	5/1/1981
SMITH	800	12/17/1980	3/27/1981	WARD	2/22/1981

3. SUMMARY

```
select  ename as "      ", sal as "      ",hiredate as "      ",
        avg(sal) over
          (order by hiredate asc range 100 preceding) as "100 -      ",
        avg(sal) over
          (order by hiredate desc range 100 preceding) as "      -100"
from emp
order by hiredate asc;
```

			100 -	-100
SMITH	800	12/17/1980	800	1216.666667
ALLEN	1600	2/20/1981	1200	2168.75
WARD	1250	2/22/1981	1216.666667	2358.333333
JONES	2975	4/2/1981	1941.666667	2758.333333
BLAKE	2850	5/1/1981	2168.75	2650
CLARK	2450	6/9/1981	2758.333333	1975
TURNER	1500	9/8/1981	1975	2340
MARTIN	1250	9/28/1981	1375	2550

KING	5000	11/17/1981	2583.333333	2562.5
JAMES	950	12/3/1981	2340	1750
FORD	3000	12/3/1981	2340	1750
MILLER	1300	1/23/1982	2562.5	1300
SCOTT	3000	4/19/1987	3000	2050
ADAMS	1100	5/23/1987	2050	1100

2_4_2. Row Windows

1. Asc

```
select  ename as "      ", sal as "      ", hiredate as "      ",
        first_value(ename) over
          (order by hiredate asc rows 5 preceding ) as "      _pre",
        first_value(hiredate) over
          (order by hiredate asc rows 5 preceding ) as "      _pre"
from emp
order by hiredate asc;
```

			_pre	_pre
SMITH	800	12/17/1980	SMITH	12/17/1980
ALLEN	1600	2/20/1981	SMITH	12/17/1980
WARD	1250	2/22/1981	SMITH	12/17/1980
JONES	2975	4/2/1981	SMITH	12/17/1980
BLAKE	2850	5/1/1981	SMITH	12/17/1980
CLARK	2450	6/9/1981	SMITH	12/17/1980
TURNER	1500	9/8/1981	ALLEN	2/20/1981
MARTIN	1250	9/28/1981	WARD	2/22/1981
KING	5000	11/17/1981	JONES	4/2/1981
JAMES	950	12/3/1981	BLAKE	5/1/1981
FORD	3000	12/3/1981	CLARK	6/9/1981
MILLER	1300	1/23/1982	TURNER	9/8/1981
SCOTT	3000	4/19/1987	MARTIN	9/28/1981
ADAMS	1100	5/23/1987	KING	11/17/1981

2. Desc

```
select  ename as "      ", sal as "      ", hiredate as "      ",
        first_value(ename) over
          (order by hiredate desc rows 5 preceding ) as "      _pre",
        first_value(hiredate) over
          (order by hiredate desc rows 5 preceding ) as "      _pre"
from emp
order by hiredate desc;
```

			_pre	_pre
ADAMS	1100	5/23/1987	ADAMS	5/23/1987
SCOTT	3000	4/19/1987	ADAMS	5/23/1987

MILLER	1300	1/23/1982	ADAMS	5/23/1987
JAMES	950	12/3/1981	ADAMS	5/23/1987
FORD	3000	12/3/1981	ADAMS	5/23/1987
KING	5000	11/17/1981	ADAMS	5/23/1987
MARTIN	1250	9/28/1981	SCOTT	4/19/1987
TURNER	1500	9/8/1981	MILLER	1/23/1982
CLARK	2450	6/9/1981	JAMES	12/3/1981
BLAKE	2850	5/1/1981	FORD	12/3/1981
JONES	2975	4/2/1981	KING	11/17/1981
WARD	1250	2/22/1981	MARTIN	9/28/1981
ALLEN	1600	2/20/1981	TURNER	9/8/1981
SMITH	800	12/17/1980	CLARK	6/9/1981

3 SUMMARY

```

select hiredate as "    ", ename as "    ", sal as "    ",
       avg(sal) over (order by hiredate asc rows 5 preceding ) as "    5 _    ",
       count(*) over (order by hiredate asc rows 5 preceding) as " bef_count",
       avg(sal) over (order by hiredate desc rows 5 preceding ) as "    5 _    ",
       count(*) over (order by hiredate desc rows 5 preceding) as " aft_count"
from emp
order by hiredate asc, ename asc;

```

			5 _	bef_count	5 _	aft_count
12/17/1980	SMITH	800	800	1	1987.5	6
2/20/1981	ALLEN	1600	1200	2	2104.166667	6
2/22/1981	WARD	1250	1216.667	3	2045.833333	6
4/2/1981	JONES	2975	1656.25	4	2670.833333	6
5/1/1981	BLAKE	2850	1895	5	2675	6
6/9/1981	CLARK	2450	1987.5	6	2358.333333	6
9/8/1981	TURNER	1500	2104.167	6	2166.666667	6
9/28/1981	MARTIN	1250	2045.833	6	2416.666667	6
11/17/1981	KING	5000	2670.833	6	2391.666667	6
12/3/1981	FORD	3000	2675	6	1870	5
12/3/1981	JAMES	950	2358.333	6	1587.5	4
1/23/1982	MILLER	1300	2166.667	6	1800	3
4/19/1987	SCOTT	3000	2416.667	6	2050	2
5/23/1987	ADAMS	1100	2391.667	6	1100	1

)
 order by hiredate asc, empno asc; SORT ANAL_FUNC
 가 .

4 Spec Windows

```
select  row_number() over (order by deptno asc,hiredate asc,empno asc)
        as "      ", --empno as "      ",
        deptno as "      ", ename as "      ",
        hiredate-100 as "startdate(-100)",hiredate as "      ",
        hiredate+100 as "enddate(+100)",
        count(*) over (partition by deptno
                        order by hiredate asc
                        range 100 preceding) as "range(-100~current)cnt",
        count(*) over (partition by deptno
                        order by hiredate asc
                        range between current row and 100 following)
        as "range(current~+100)cnt",
        count(*) over (partition by deptno
                        order by hiredate asc
                        range between 100 preceding and 100 following )
        as "range(-100~+100)_cnt",
        count(*) over (partition by deptno
                        order by hiredate asc
                        rows 2 preceding)      as "row2 pre_cnt"
from emp
order by deptno,hiredate,empno;
```

			Startdate (-100)		Enddate (+100)	Range (-100~ current)cnt	Range (current~ +100)cnt	Range (-100~ +100)cnt	Row2 pre_cnt
1	10	CLARK	3/1/1981	6/9/1981	9/17/1981	1	1	1	1
2	10	KING	8/9/1981	11/17/1981	2/25/1982	1	2	2	2
3	10	MILLER	10/15/1981	1/23/1982	5/3/1982	2	1	2	3
4	20	SMITH	9/8/1980	12/17/1980	3/27/1981	1	1	1	1
5	20	JONES	12/23/1980	4/2/1981	7/11/1981	1	1	1	2
6	20	FORD	8/25/1981	12/3/1981	3/13/1982	1	1	1	3
7	20	SCOTT	1/9/1987	4/19/1987	7/28/1987	1	2	2	3
8	20	ADAMS	2/12/1987	5/23/1987	8/31/1987	2	1	2	3
9	30	ALLEN	11/12/1980	2/20/1981	5/31/1981	1	3	3	1

10	30	WARD	11/14/1980	2/22/1981	6/2/1981	2	2	3	2
11	30	BLAKE	1/21/1981	5/1/1981	8/9/1981	3	1	3	3
12	30	TURNER	5/31/1981	9/8/1981	12/17/1981	1	3	3	3
13	30	MARTIN	6/20/1981	9/28/1981	1/6/1982	2	2	3	3
14	30	JAMES	8/25/1981	12/3/1981	3/13/1982	3	1	3	3

```

select  empno as "      ", ename as "      ",
        hiredate-100 as "startdate(-100)",
        hiredate as "      ",
        hiredate+100 as "enddate(+100)",
        first_value(ename) over
            (order by hiredate asc
             range between 100 preceding and 100 following )
        as "range(-100~+100)_first",
        last_value(ename) over
            (order by hiredate asc
             range between 100 preceding and 100 following )
        as "range(-100~+100)_last"
from emp
order by hiredate, empno;

```

		Startdate (-100)		Enddate (+100)	range(-100~+100) first	range(-100~+100) last
7369	SMITH	9/8/1980	12/17/1980	3/27/1981	SMITH	WARD
7499	ALLEN	11/12/1980	2/20/1981	5/31/1981	SMITH	BLAKE
7521	WARD	11/14/1980	2/22/1981	6/2/1981	SMITH	BLAKE
7566	JONES	12/23/1980	4/2/1981	7/11/1981	ALLEN	CLARK
7698	BLAKE	1/21/1981	5/1/1981	8/9/1981	ALLEN	CLARK
7782	CLARK	3/1/1981	6/9/1981	9/17/1981	JONES	TURNER
7844	TURNER	5/31/1981	9/8/1981	12/17/1981	CLARK	JAMES
7654	MARTIN	6/20/1981	9/28/1981	1/6/1982	TURNER	JAMES
7839	KING	8/9/1981	11/17/1981	2/25/1982	TURNER	MILLER
7900	JAMES	8/25/1981	12/3/1981	3/13/1982	TURNER	MILLER
7902	FORD	8/25/1981	12/3/1981	3/13/1982	TURNER	MILLER
7934	MILLER	10/15/1981	1/23/1982	5/3/1982	KING	MILLER
7788	SCOTT	1/9/1987	4/19/1987	7/28/1987	SCOTT	ADAMS
7876	ADAMS	2/12/1987	5/23/1987	8/31/1987	SCOTT	ADAMS

3. Functions

26

(aggregate function)

avg , sum

가

1.AVG(<distinct | all> expression)

Group and window expression

Distinct

```
SELECT deptno, ename, hiredate, sal,
       AVG(sal) OVER (PARTITION BY deptno ORDER BY hiredate
                     ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING)
       AS "      rows      "
FROM emp;
```

DEPTNO	ENAME	HIREDATE	SAL	rows
10	CLARK	6/9/1981	2450	3725
10	KING	11/17/1981	5000	2916.666667
10	MILLER	1/23/1982	1300	3150
20	SMITH	12/17/1980	800	1887.5
20	JONES	4/2/1981	2975	2258.333333
20	FORD	12/3/1981	3000	2991.666667
20	SCOTT	4/19/1987	3000	2366.666667
20	ADAMS	5/23/1987	1100	2050
30	ALLEN	2/20/1981	1600	1425
30	WARD	2/22/1981	1250	1900
30	BLAKE	5/1/1981	2850	1866.666667
30	TURNER	9/8/1981	1500	1866.666667
30	MARTIN	9/28/1981	1250	1233.333333
30	JAMES	12/3/1981	950	1100

2.COUNT (<distinct> <*> <expression>)

* non-null , count rows
 expression count expression not-null 가
 rows DISTINCT

```
SELECT ename, empno, sal,
```

```

COUNT(*) OVER (ORDER BY sal asc
                  RANGE BETWEEN 50 PRECEDING    AND 150 FOLLOWING)
AS "    50-    150    "
FROM emp;

```

ENAME	EMPNO	SAL	50 - 150
SMITH	7369	800	2
JAMES	7900	950	2
ADAMS	7876	1100	3
WARD	7521	1250	3
MARTIN	7654	1250	3
MILLER	7934	1300	3
TURNER	7844	1500	2
ALLEN	7499	1600	1
CLARK	7782	2450	1
BLAKE	7698	2850	4
JONES	7566	2975	3
SCOTT	7788	3000	3
FORD	7902	3000	3
KING	7839	5000	1

3.MAX (expression)

expression

```

SELECT deptno,empno, ename, sal,
       MIN(sal) OVER ( PARTITION BY deptno ORDER BY empno asc) AS dept_mix,
       MAX(sal) OVER ( PARTITION BY deptno ORDER BY empno asc) AS dept_max
FROM emp;

```

DEPTNO	EMPNO	ENAME	SAL	DEPT_MIX	DEPT_MAX
10	7782	CLARK	2450	2450	5000
10	7839	KING	5000	2450	5000
10	7934	MILLER	1300	1300	5000
20	7369	SMITH	800	800	3000
20	7566	JONES	2975	800	3000
20	7788	SCOTT	3000	800	3000
20	7876	ADAMS	1100	800	3000

20	7902	FORD	3000	800	3000
30	7499	ALLEN	1600	1600	2850
30	7521	WARD	1250	1250	2850
30	7654	MARTIN	1250	1250	2850
30	7698	BLAKE	2850	1250	2850
30	7844	TURNER	1500	1250	2850
30	7900	JAMES	950	950	2850

4.MIN (expression)

expression

```
SELECT deptno, empno, ename, hiredate, sal,
       MIN(sal) OVER(PARTITION BY deptno ORDER BY hiredate
                     RANGE UNBOUNDED PRECEDING) as "      "
FROM emp;
```

DEPTNO	EMPNO	ENAME	HIREDATE	SAL	—
10	7782	CLARK	6/9/1981	2450	2450
10	7839	KING	11/17/1981	5000	2450
10	7934	MILLER	1/23/1982	1300	1300
20	7369	SMITH	12/17/1980	800	800
20	7566	JONES	4/2/1981	2975	800
20	7902	FORD	12/3/1981	3000	800
20	7788	SCOTT	4/19/1987	3000	800
20	7876	ADAMS	5/23/1987	1100	800
30	7499	ALLEN	2/20/1981	1600	1600
30	7521	WARD	2/22/1981	1250	1250
30	7698	BLAKE	5/1/1981	2850	1250
30	7844	TURNER	9/8/1981	1500	1250
30	7654	MARTIN	9/28/1981	1250	1250
30	7900	JAMES	12/3/1981	950	950

5.RATIO_TO_REPORT (expression)

Expression / SUM(expression) row

```

SELECT  ename, sal,
        RATIO_TO_REPORT (sal) OVER () AS "RATIO_TO_REPORT (sal)"
FROM emp;

```

ENAME	SAL	RATIO_TO_REPORT (sal)
SMITH	800	0.027562446
ALLEN	1600	0.055124892
WARD	1250	0.043066322
JONES	2975	0.102497847
MARTIN	1250	0.043066322
BLAKE	2850	0.098191214
CLARK	2450	0.084409991
SCOTT	3000	0.103359173
KING	5000	0.172265289
TURNER	1500	0.051679587
ADAMS	1100	0.037898363
JAMES	950	0.032730405
FORD	3000	0.103359173
MILLER	1300	0.044788975

6.SUM (expression)

expression

```

SELECT mgr, ename, sal,
       SUM(sal) OVER ( PARTITION BY mgr ORDER BY sal
                       RANGE UNBOUNDED PRECEDING) AS l_csum
FROM emp;
ORDER BY sal ,ename asc      가
order by

```

MGR	ENAME	SAL	L_CSUM
7566	SCOTT	3000	6000
7566	FORD	3000	6000
7698	JAMES	950	950
7698	WARD	1250	3450
7698	MARTIN	1250	3450
7698	TURNER	1500	4950

7698	ALLEN	1600	6550
7782	MILLER	1300	1300
7788	ADAMS	1100	1100
7839	CLARK	2450	2450
7839	BLAKE	2850	5300
7839	JONES	2975	8275
7902	SMITH	800	800
	KING	5000	5000

7.FIRST_VALUE(expression)

(FIRST VALUE)

```

SELECT deptno, ename, sal,
       FIRST_VALUE(ename) OVER
       ( ORDER BY deptno asc,sal asc, ename asc  ROWS UNBOUNDED PRECEDING)
       AS FIRST_VALUE,
       LAST_VALUE(ename) OVER
       ( ORDER BY deptno asc,sal asc, ename asc  ROWS BETWEEN CURRENT ROW AND UNBOUNDED
       FOLLOWING) AS LAST_VALUE
FROM emp
ORDER BY deptno asc, sal asc,ename asc

```

DEPTNO	ENAME	SAL	FIRST_VALUE	LAST_VALUE
10	CLARK	2450	MILLER	BLAKE
10	KING	5000	MILLER	BLAKE
10	MILLER	1300	MILLER	BLAKE
20	ADAMS	1100	MILLER	BLAKE
20	FORD	3000	MILLER	BLAKE
20	JONES	2975	MILLER	BLAKE
20	SCOTT	3000	MILLER	BLAKE
20	SMITH	800	MILLER	BLAKE
30	ALLEN	1600	MILLER	BLAKE
30	BLAKE	2850	MILLER	BLAKE
30	JAMES	950	MILLER	BLAKE
30	MARTIN	1250	MILLER	BLAKE
30	TURNER	1500	MILLER	BLAKE
30	WARD	1250	MILLER	BLAKE

8. LAST_VALUE(expression)

(LAST VALUE)

9. LAG (expression, <offset>, <default>)

Self-join resultset rows 가
 current row rows reference
 row 가 rows select
 row LEAD

Offset : default 1

Default: index가 window

row

SELECT ename, hiredate, sal,

LAG(sal, 1, 0) OVER (ORDER BY hiredate) as prev_sal

FROM emp

WHERE job = 'SALESMAN';

ENAME	HIREDATE	SAL	PREV_SAL
ALLEN	2/20/1981	1600	0
WARD	2/22/1981	1250	1600
TURNER	9/8/1981	1500	1250
MARTIN	9/28/1981	1250	1500

10. LEAD (expression, <offset>, <default>)

LAG row row

Offset : default 1

Default: index가 window

)

SELECT ename, hiredate,

LEAD(hiredate, 1,2000-01-01) OVER (ORDER BY hiredate) AS next_hire_date

FROM emp;

ENAME	HIREDATE	NEXT_HIRE_DATE
SMITH	12/17/1980	2/20/1981

ALLEN	2/20/1981	2/22/1981
WARD	2/22/1981	4/2/1981
JONES	4/2/1981	5/1/1981
BLAKE	5/1/1981	6/9/1981
CLARK	6/9/1981	9/8/1981
TURNER	9/8/1981	9/28/1981
MARTIN	9/28/1981	11/17/1981
KING	11/17/1981	12/3/1981
JAMES	12/3/1981	12/3/1981
FORD	12/3/1981	1/23/1982
MILLER	1/23/1982	4/19/1987
SCOTT	4/19/1987	5/23/1987
ADAMS	5/23/1987	1/1/2000

11. NTILE (expression)

expression

```
SELECT ename, sal, NTILE(4) OVER (ORDER BY sal DESC) AS quartile
FROM emp;
```

ENAME	SAL	QUARTILE
KING	5000	1
SCOTT	3000	1
FORD	3000	1
JONES	2975	1
BLAKE	2850	2
CLARK	2450	2
ALLEN	1600	2
TURNER	1500	2
MILLER	1300	3
WARD	1250	3
MARTIN	1250	3
ADAMS	1100	4
JAMES	950	4
SMITH	800	4

```
SELECT ename, sal, NTILE(10) OVER (ORDER BY sal DESC) AS quartile
FROM emp;
```

ENAME	SAL	QUARTILE
KING	5000	1
SCOTT	3000	1
FORD	3000	2
JONES	2975	2
BLAKE	2850	3
CLARK	2450	3
ALLEN	1600	4
TURNER	1500	4
MILLER	1300	5
WARD	1250	6
MARTIN	1250	7
ADAMS	1100	8
JAMES	950	9
SMITH	800	10

```
SELECT ename, sal, NTILE(100) OVER (ORDER BY sal DESC) AS quartile
FROM emp;
```

ENAME	SAL	QUARTILE
KING	5000	1
SCOTT	3000	2
FORD	3000	3
JONES	2975	4
BLAKE	2850	5
CLARK	2450	6
ALLEN	1600	7
TURNER	1500	8
MILLER	1300	9
WARD	1250	10
MARTIN	1250	11
ADAMS	1100	12
JAMES	950	13
SMITH	800	14

12. RANK()

ORDER BY expression rows row
 . order by sort , row 1
 assign .
 rows . 2 row가 ,
 Skip .
 2 row가 1 row 2 가 3 assign .
 Skip DENSE_RANK .

```
SELECT deptno, ename, sal, comm,
       RANK() OVER (PARTITION BY deptno ORDER BY sal DESC, comm) as rk,
       DENSE_RANK() OVER (PARTITION BY deptno ORDER BY sal DESC, comm) as drk
FROM emp;
```

DEPTNO	ENAME	SAL	COMM	RK	DRK
10	KING	5000		1	1
10	CLARK	2450		2	2
10	MILLER	1300		3	3
20	SCOTT	3000		1	1
20	FORD	3000		1	1
20	JONES	2975		3	2
20	ADAMS	1100		4	3
20	SMITH	800		5	4
30	BLAKE	2850		1	1
30	ALLEN	1600	300	2	2
30	TURNER	1500	0	3	3
30	WARD	1250	500	4	4
30	MARTIN	1250	1400	5	5
30	JAMES	950		6	6

13. DENSE_RANK()

ORDER BY expression rows row
 . order by sort , row 1
 assign .
 ORDER BY expression 가 .
 rows GAPS 가 .
 RANK () .

14. CUME_DIST()

```

SELECT job, ename, sal, CUME_DIST()
  OVER (PARTITION BY job ORDER BY sal) AS cume_dist
FROM emp;

```

JOB	ENAME	SAL		CUME_DIST
ANALYST	SCOTT	3000	1/1	1
ANALYST	FORD	3000	1/1	1
CLERK	SMITH	800	1/4	0.25
CLERK	JAMES	950	2/4	0.5
CLERK	ADAMS	1100	3/4	0.75
CLERK	MILLER	1300	4/4	1
MANAGER	CLARK	2450	1/3	0.33
MANAGER	BLAKE	2850	2/3	0.66
MANAGER	JONES	2975	3/3	1
PRESIDENT	KING	5000	1/1	1
SALESMAN	WARD	1250	2/4	0.5
SALESMAN	MARTIN	1250	2/4	0.5
SALESMAN	TURNER	1500	3/4	0.75
SALESMAN	ALLEN	1600	4/4	1

15. PERCENT_RANK()

CUME_DIST .
 0 1 .
 row

```

SELECT deptno, ename, sal,
  PERCENT_RANK() OVER (PARTITION BY deptno ORDER BY sal DESC) AS pr
FROM emp;

```

DEPTNO	ENAME	SAL	PR
10	KING	5000	0
10	CLARK	2450	0.5
10	MILLER	1300	1
20	SCOTT	3000	0
20	FORD	3000	0
20	JONES	2975	0.5

20	ADAMS	1100	0.75
20	SMITH	800	1
30	BLAKE	2850	0
30	ALLEN	1600	0.2
30	TURNER	1500	0.4
30	WARD	1250	0.6
30	MARTIN	1250	0.6
30	JAMES	950	1

#

16. CORR (expression, expression) : expression .

17. COVAR_POP (expression, expression) : expression .

18. COVAR_SAMP (expression, expression) : expression .

19. STDDEV (expression) : CURRENT ROW

20. STDDEV_POP (expression) :

, VAR_POP

21. STDDEV_SAMP (expression) :

, VAR_SAMP

22. VAR_POP (expression) : NULL NOT -NULL

23. VAR_SAMP (expression) : NULL NOT -NULL

24. VARIANCE (expression) : expression

expression row 가 1 0

expression row > 1 VAR_SAMP

25. REGR_XXXXXXX (expression, expression) : . 9 가 가 .

: !