Oracle数据库中的函数有多种，比如单行函数、聚合函数、对象引用函数、模型函数、OLAP函数等。本篇将详细介绍Oracle数据库中的分析函数。

1 分析函数概述

所谓分析函数，是基于一组数据行计算聚合值，其与聚合函数的不同之处在于，它为每一组返回多个数据行。一组数据行称为一个窗口，由analytic_clause子句进行定义，对于每一行，定义一个行移动窗口，窗口确定用于为当前行执行计算的行的范围，它的大小可以基于物理行或逻辑间隔（如时间）。

2 分析函数语法

analytic_function::=

说明：

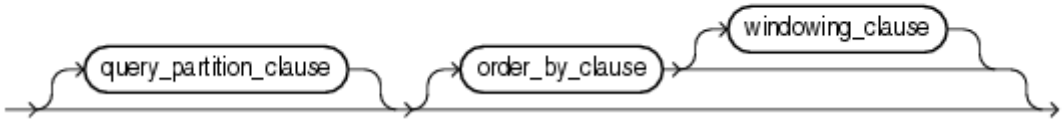
analytic_function

指出分析函数的名称；

arguments

分析函数的参数，参数数量在0个到3个之间，该参数类型可以是任何数值数据类型，或者任何可以隐式转换为数值数据类型的非数值类型。

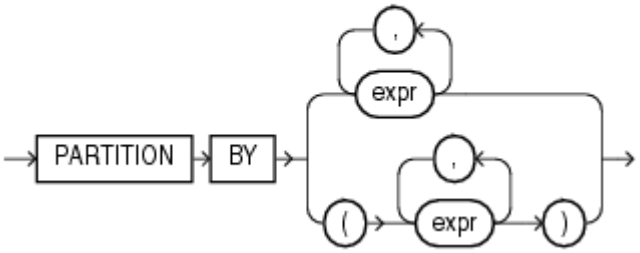
analytic_clause::=



analytic_clause

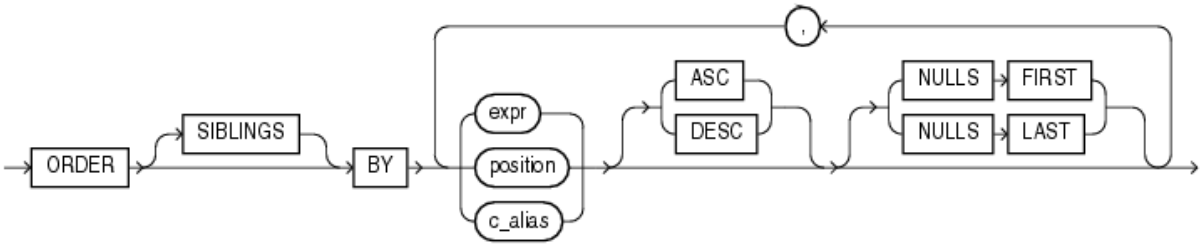
使用OVER analytic_clause子句展示在查询结果集上所进行的函数操作，该子句是在FROM、WHERE、GROUP BY和HAVING子句后进行计算的。

query_partition_clause::=



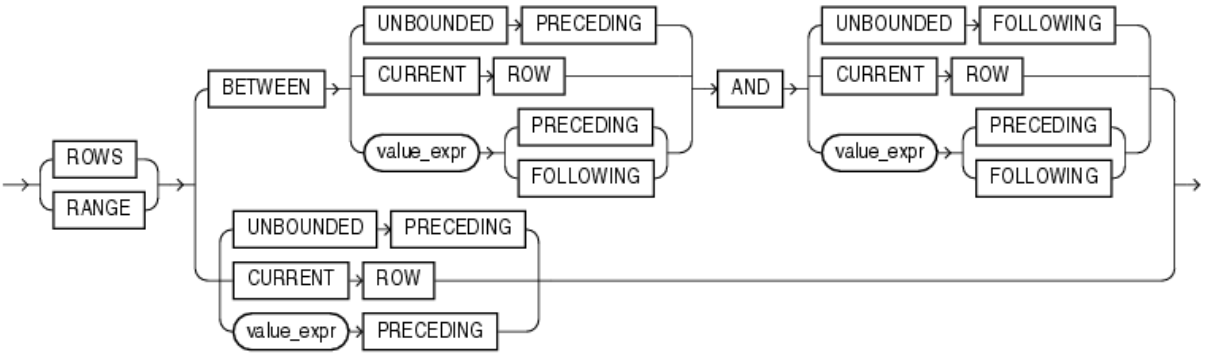
使用PARTITION BY子句可以基于一个或多个expr将查询结果集划分为多个组，如果忽略该子句，则函数会将整个查询结果集当做一个组。

order_by_clause::=



order_by_clause子句用于指定数据是在一个分组里是如何进行排序的。

windowing_clause ::=



3 分析函数类型

Oracle有以下类型的分析函数，其中，带有*号的函数可以使用上面的完全语句，包括窗口子句。

- AVG *
- COPR *
- COUNT *
- COVAR_POP *
- COVAR_SAMP *
- CUME_DIST
- DENSE_RANK
- FIRST

- FIRST_VALUE *
- LAG
- LAST
- LAST_VALUE *
- LEAD
- LISTAGG
- MAX *
- MEDIAN
- MIN *
- NTH_VALUE *
- NTILE
- PERCENT_RANK
- PERCENTILE_COUNT
- PERCENTILE_DISC
- RANK
- RATIO_TO_REPORT
- REGR_(Linear Regression) Functions *
- ROW_NUMBER
- STDDEV *
- STDDEV_POP *
- STDDEV_SAMP *
- SUM *
- VAR_POP *
- VAR_SAMP *
- VARIANCE *

4 分析函数详解

4.1 演示数据库版本

SQL> select *from product_component_version where product like 'Oracle%';

PRODUCT VERSION STATUS

---------- ---------- ----------

4.2 AVG函数

1）语法结构

Syntax

AVG函数用于返回expr的平均值，如果指定DISTINCT，那么只能指定analytic_clause中的query_partition_clause子句，不能指定order_by_clause andwindowing_clause子句。

2）示例

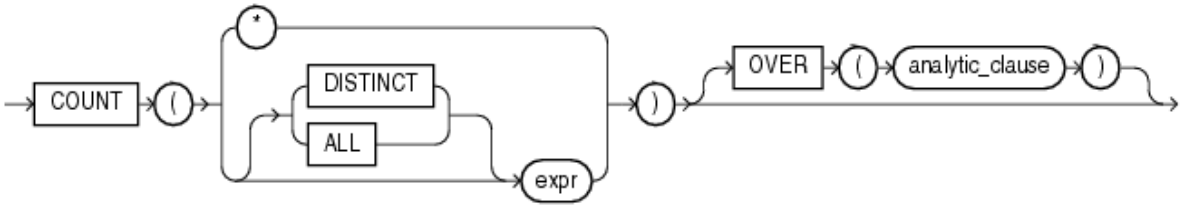
```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       deptno,
       sal,
       round(AVG(sal) over(), 2) avg_all_sal,
       round(AVG(sal) over(PARTITION BY deptno), 2) avg_dept_sal,
       round(AVG(sal) over(PARTITION BY deptno ORDER BY empno
                 rows BETWEEN 1 preceding AND 1 following),
             2) avg_sal1,
       round(AVG(sal) over(PARTITION BY deptno ORDER BY empno
                 rows BETWEEN CURRENT ROW AND 1 following),
             2) avg_sal2
  FROM  emp;
```

```
     EMPNO ENAME      JOB              MGR HIREDATE              DEPTNO        SAL AVG_ALL_SAL AVG_DEPT_SAL   AVG_SAL1   AVG_SAL2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ----------- ------------ ---------- ----------
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00         10       2450     2073.21      2916.67       3725       3725
      7839 KING       PRESIDENT            1981-11-17 00:00:00         10       5000     2073.21      2916.67    2916.67       3150
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00         10       1300     2073.21      2916.67       3150       1300
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00         20        800     2073.21         2175     1887.5     1887.5
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00         20       2975     2073.21         2175    2258.33     2987.5
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00         20       3000     2073.21         2175    2358.33       2050
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00         20       1100     2073.21         2175    2366.67       2050
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00         20       3000     2073.21         2175       2050       3000
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00         30       1600     2073.21      1566.67       1425       1425
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00         30       1250     2073.21      1566.67    1366.67       1250
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00         30       1250     2073.21      1566.67    1783.33       2050

     EMPNO ENAME      JOB              MGR HIREDATE              DEPTNO        SAL AVG_ALL_SAL AVG_DEPT_SAL   AVG_SAL1   AVG_SAL2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ----------- ------------ ---------- ----------
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00         30       2850     2073.21      1566.67    1866.67       2175
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00         30       1500     2073.21      1566.67    1766.67       1225
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00         30        950     2073.21      1566.67       1225        950

14 rows selected.
```

4.3 COUNT

1）语法结构

Syntax



COUNT返回查询的行数，如果指定DISTINCT，那么只能指定analytic_clause中的query_partition_clause子句，不能指定order_by_clause andwindowing_clause子句。如果指定expr，COUNT返回expr非空的行数，如果指定*，则返回所有行数，包括重复行和空行。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       deptno,
       sal,
       comm,
       COUNT(1) over() qty1,
       COUNT(*) over() qty2,
       COUNT(comm) over() qty3,
       COUNT(DISTINCT deptno) over() qty4,
       COUNT(1) over(PARTITION BY deptno) qty5
```

FROM emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	DEPTNO	SAL	COMM	QTY1	QTY2	QTY3	QTY4	QTY5
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	10	2450		14	14	4	3	3
7839	KING	PRESIDENT		1981-11-17 00:00:00	10	5000		14	14	4	3	3
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	10	1300		14	14	4	3	3
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	20	2975		14	14	4	3	5
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	20	3000		14	14	4	3	5
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	20	1100		14	14	4	3	5
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	20	800		14	14	4	3	5
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	20	3000		14	14	4	3	5
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	30	1250	500	14	14	4	3	6
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	30	1500	0	14	14	4	3	6
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	30	1600	300	14	14	4	3	6

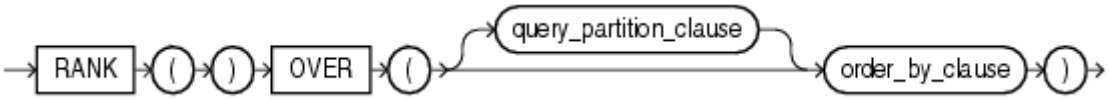
EMPNO	ENAME	JOB	MGR	HIREDATE	DEPTNO	SAL	COMM	QTY1	QTY2	QTY3	QTY4	QTY5
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	30	950		14	14	4	3	6
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	30	2850		14	14	4	3	6
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	30	1250	1400	14	14	4	3	6

14 rows selected.

4.4 RANK、DENSE_RANK、ROW_NUMBER

1）语法结构

***rank_analytic*::=**



***dense_rank_analytic*::=**

Syntax

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
       --row_number() over(ORDER BY sal) rn1,
       row_number() over(PARTITION BY deptno ORDER BY sal) rn2,
       --rank()over(order by sal) rn3
```

rank() over(PARTITION BY deptno ORDER BY sal) r4,

dense_rank() over(PARTITION BY deptno ORDER BY sal) r5

FROM emp;

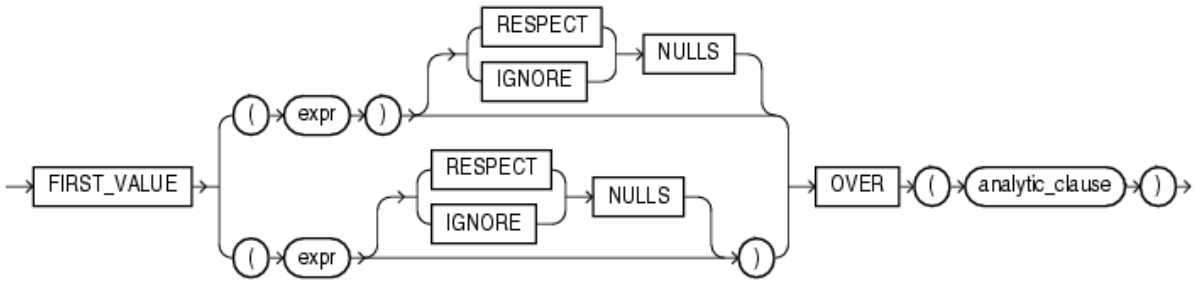
```
     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO        RN2         R4         R5
---------- ---------- --------- ---------- ------------------- ---------- ---------- ---------- ---------- ----------
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300         10          1          1          1
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450         10          2          2          2
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000         10          3          3          3
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800         20          1          1          1
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100         20          2          2          2
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975         20          3          3          3
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000         20          4          4          4
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000         20          5          4          4
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950         30          1          1          1
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250         30          2          2          2
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250         30          3          2          2

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO        RN2         R4         R5
---------- ---------- --------- ---------- ------------------- ---------- ---------- ---------- ---------- ----------
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500         30          4          4          3
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600         30          5          5          4
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850         30          6          6          5

14 rows selected.
```

> 这三个函数主要用于进行排序，ROW_NUMBER函数排序的结果是连续的，而DENSE函数排序会出现跳号，DENSE_RANK函数排序，不会出现跳号。

4.5 FIRST_VALUE

1）语法结构

Syntax



FIRST_VALUE用于返回在一个排序结果集中的第一个值。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
       first_value(sal) over(PARTITION BY deptno ORDER BY sal) sal,
       first_value(sal) over(PARTITION BY deptno ORDER BY sal DESC) sal2
```

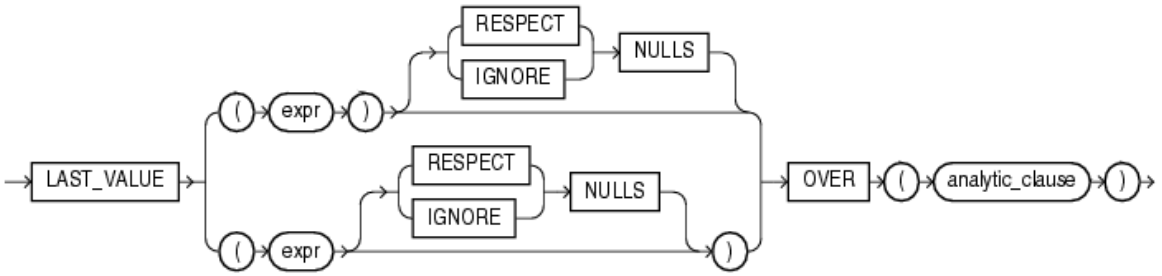
FROM emp;

```
     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO        SAL       SAL2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ---------- ----------
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300         10       1300       5000
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450         10       1300       5000
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000         10       1300       5000
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800         20        800       3000
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100         20        800       3000
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975         20        800       3000
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000         20        800       3000
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000         20        800       3000
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950         30        950       2850
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250         30        950       2850
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250         30        950       2850

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO        SAL       SAL2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ---------- ----------
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500         30        950       2850
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600         30        950       2850
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850         30        950       2850

14 rows selected.
```

4.6 LAST_VALUE

1）语法结构

Syntax



对于该函数，如果忽略了analytic_clause子句中的windowing_clause子句，默认是使用RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW，使用默认值有时可能返回不是预期的结果，因为LAST VALUE是窗口中的最后一个值，它不是固定的，而是随着当前行的改变而改变。为了得到与其的结果，可以使用RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING窗口，也可以使用RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING窗口。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
```

last_value(sal) over(PARTITION BY deptno ORDER BY sal) sal1,

last_value(sal) over(PARTITION BY deptno ORDER BY sal RANGE BETWEEN unbounded preceding AND CURRENT ROW) sal2,

last_value(sal) over(PARTITION BY deptno ORDER BY sal RANGE BETWEEN unbounded preceding AND unbounded following) sal3,

last_value(sal) over(PARTITION BY deptno ORDER BY sal RANGE BETWEEN CURRENT ROW AND unbounded following) sal4,

last_value(sal) over(PARTITION BY deptno ORDER BY sal DESC RANGE BETWEEN unbounded preceding AND unbounded following) sal5,

last_value(sal) over(PARTITION BY deptno ORDER BY sal rows BETWEEN unbounded preceding AND unbounded following) sal6

FROM emp;

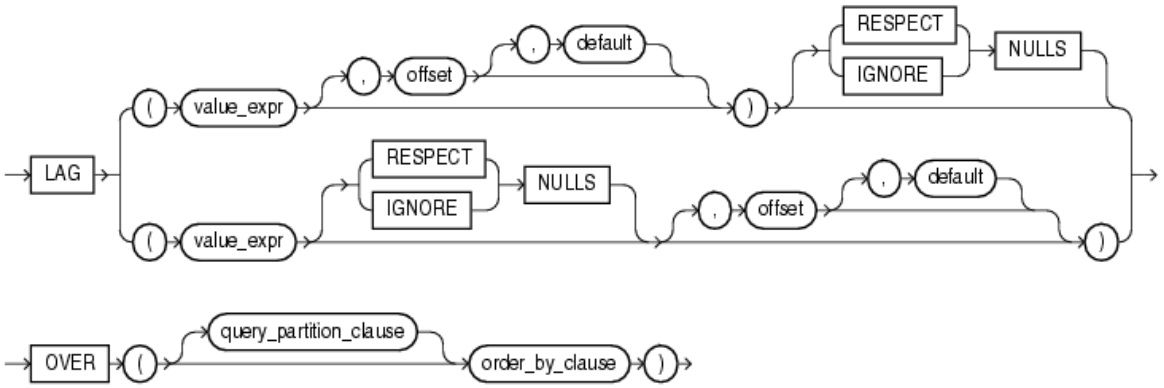
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3	SAL4	SAL5	SAL6
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	10	1300	1300	5000	5000	1300	5000
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	10	2450	2450	5000	5000	1300	5000
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000	10	5000	5000	5000	5000	1300	5000
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	20	800	800	3000	3000	800	3000
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	20	1100	1100	3000	3000	800	3000
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	20	2975	2975	3000	3000	800	3000
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	20	3000	3000	3000	3000	800	3000
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	20	3000	3000	3000	3000	800	3000
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	30	950	950	2850	2850	950	2850
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	30	1250	1250	2850	2850	950	2850
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	30	1250	1250	2850	2850	950	2850
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3	SAL4	SAL5	SAL6
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	30	1500	1500	2850	2850	950	2850
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	30	1600	1600	2850	2850	950	2850
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	30	2850	2850	2850	2850	950	2850

14 rows selected.

4.7 LAG

1）语法结构

Syntax



该函数提供了在不使用自连接的情况下同时访问表的多行的能力，即将数据行根据偏移量靠后，offset参数指定偏移量，默认值为1，default参数指定超过窗口边界的默认值，没有指定，则为NULL。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
       lag(sal) over(PARTITION BY deptno ORDER BY sal) sal1,
       lag(sal, 1) over(PARTITION BY deptno ORDER BY sal) sal2,
       lag(sal, 2, 0) over(PARTITION BY deptno ORDER BY sal) sal3
  FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	10			0
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	10	1300	1300	0
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000	10	2450	2450	1300
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	20			0
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	20	800	800	0
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	20	1100	1100	800
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	20	2975	2975	1100
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	20	3000	3000	2975
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	30			0
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	30	950	950	0
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	30	1250	1250	950

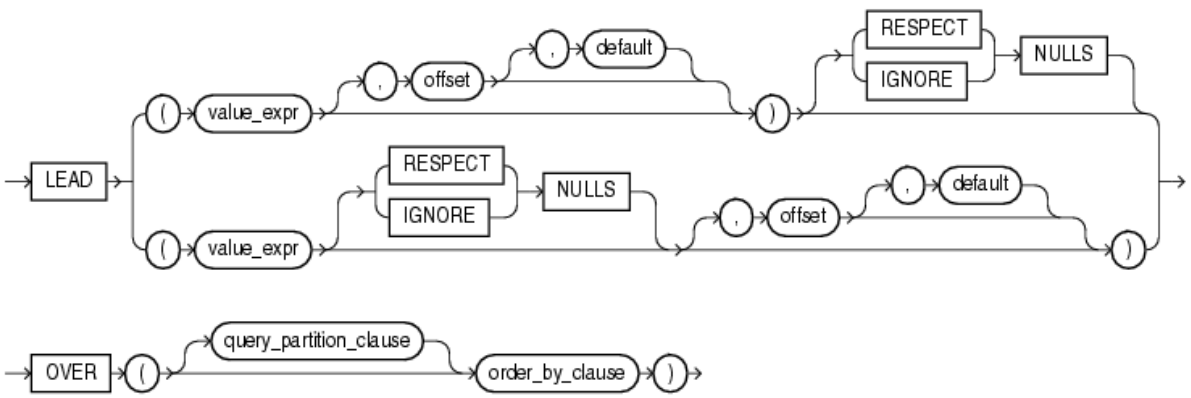
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	30	1250	1250	1250
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	30	1500	1500	1250
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	30	1600	1600	1500

14 rows selected.

4.8 LEAD

1）语法结构

Syntax



该函数和LAG函数功能相反，主要用于将行数据提前。

2）示例

```
SQL> SELECT empno,
```

```
    ename,
    job,
    mgr,
    hiredate,
    sal,
    deptno,
    lead(sal) over(PARTITION BY deptno ORDER BY sal) sal1,
    lead(sal, 1) over(PARTITION BY deptno ORDER BY sal) sal2,
    lead(sal, 2, 0) over(PARTITION BY deptno ORDER BY sal) sal3
FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	10	2450	2450	5000
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	10	5000	5000	0
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000	10			0
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	20	1100	1100	2975
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	20	2975	2975	3000
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	20	3000	3000	3000
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	20	3000	3000	0
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	20			0
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	30	1250	1250	1250
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	30	1250	1250	1500
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	30	1500	1500	1600

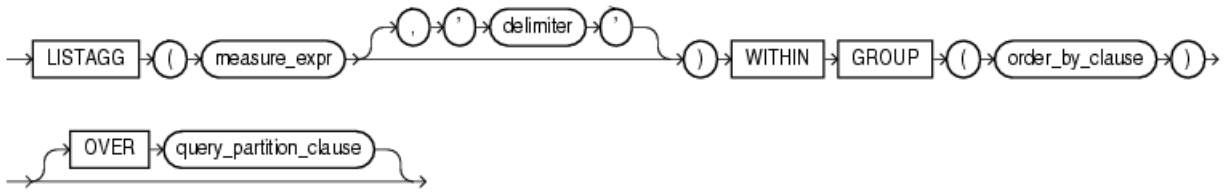
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	30	1600	1600	2850
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	30	2850	2850	0
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	30			0

14 rows selected.

4.9 LISTAGG

1）语法结构

Syntax



该函数用于在分组内对数据进行排序，然后将对应的值进行拼接，参数delimilter用于指定拼接符。

2）示例

```
SQL> SELECT empno,
    ename,
    job,
    mgr,
```

hiredate,
sal,
deptno,
listagg(ename, ',') within GROUP(ORDER BY empno) over(PARTITION BY deptno) name1
FROM emp;

```
     EMPNO ENAME      JOB              MGR HIREDATE                SAL     DEPTNO NAME1
---------- ---------- --------- ---------- ------------------ ---------- ---------- ------------------------------------
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450         10 CLARK,KING,MILLER
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000         10 CLARK,KING,MILLER
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300         10 CLARK,KING,MILLER
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800         20 SMITH,JONES,SCOTT,ADAMS,FORD
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975         20 SMITH,JONES,SCOTT,ADAMS,FORD
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000         20 SMITH,JONES,SCOTT,ADAMS,FORD
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100         20 SMITH,JONES,SCOTT,ADAMS,FORD
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000         20 SMITH,JONES,SCOTT,ADAMS,FORD
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES

     EMPNO ENAME      JOB              MGR HIREDATE                SAL     DEPTNO NAME1
---------- ---------- --------- ---------- ------------------ ---------- ---------- ------------------------------------
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950         30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES

14 rows selected.
```

思考：如果拼接的列数据类型为varchar2，则有无长度限制，如果超过了该怎么处理？

```
SQL> SELECT deptno,
       dbms_lob.substr(substr(rtrim(xmlcast(xmlagg(xmlelement(e,
                                                              ename || ',')
                                                   ORDER BY empno) AS CLOB),
                                    ','),
                              1,
                              100)) NAME
  FROM emp
 GROUP BY deptno;
```

```
    DEPTNO NAME
---------- ----------------------------------------
        10 CLARK,KING,MILLER
        20 SMITH,JONES,SCOTT,ADAMS,FORD
        30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

4.10 MAX、MIN、SUM

1）语法结构

Syntax

Syntax

Syntax

这三个函数主要用于在分组内求最大值、最小值以及求和。

2）示例

```
SQL> SELECT empno,
        ename,
        job,
        mgr,
        hiredate,
        sal,
        deptno,
        MAX(sal) over() sal1,
        MAX(sal) over(PARTITION BY deptno) sal2,
        MAX(sal) over(PARTITION BY deptno ORDER BY empno rows BETWEEN
unbounded preceding AND 1 following) sal3,
        MIN(sal) over() sal4,
        MIN(sal) over(PARTITION BY deptno) sal5,
        MIN(sal) over(PARTITION BY deptno ORDER BY empno rows BETWEEN
unbounded preceding AND 1 following) sal6,
        SUM(sal) over() sal7,
        SUM(sal) over(PARTITION BY deptno) sal8,
```

SUM(sal) over(PARTITION BY deptno ORDER BY empno rows BETWEEN unbounded preceding AND 1 following) sal9
FROM emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3	SAL4	SAL5	SAL6	SAL7	SAL8	SAL9
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	10	5000	5000	5000	800	1300	2450	29025	8750	7450
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000	10	5000	5000	5000	800	1300	1300	29025	8750	8750
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	10	5000	5000	5000	800	1300	1300	29025	8750	8750
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	20	5000	3000	2975	800	800	800	29025	10875	3775
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	20	5000	3000	3000	800	800	800	29025	10875	6775
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	20	5000	3000	3000	800	800	800	29025	10875	7875
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	20	5000	3000	3000	800	800	800	29025	10875	10875
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	20	5000	3000	3000	800	800	800	29025	10875	10875
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	30	5000	2850	1600	800	950	1250	29025	9400	2850
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	30	5000	2850	1600	800	950	1250	29025	9400	4100
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	30	5000	2850	2850	800	950	1250	29025	9400	6950

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	SAL1	SAL2	SAL3	SAL4	SAL5	SAL6	SAL7	SAL8	SAL9
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	30	5000	2850	2850	800	950	1250	29025	9400	8450
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	30	5000	2850	2850	800	950	950	29025	9400	9400
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	30	5000	2850	2850	800	950	950	29025	9400	9400

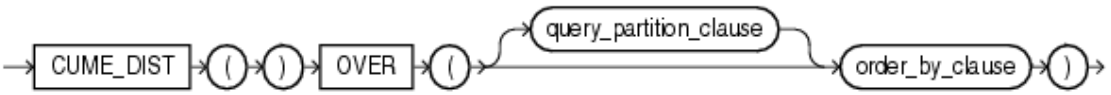
14 rows selected.

4.11 CUME_DIST

1）语法结构

Analytic Syntax

cume_dist_analytic::=



该函数用于计算一个值在一组值中的累积分布，返回的值的范围是0到1。计算逻辑为：在一组值中某一个值的相对位置，即对于某一行R，总行数为N，值为R/N。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
       cume_dist() over(ORDER BY sal) p1,
       cume_dist() over(PARTITION BY deptno ORDER BY sal) p2
  FROM emp
 ORDER BY deptno, sal;
```

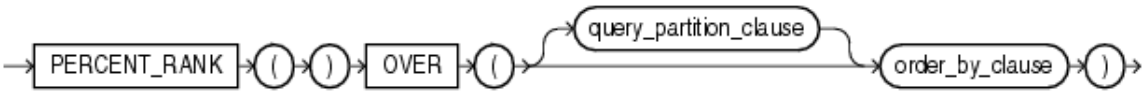
```
     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO         P1         P2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ---------- ----------
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300         10 .428571429 .333333333
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450         10 .642857143 .666666667
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000         10          1          1
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800         20 .071428571         .2
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100         20 .214285714         .4
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975         20 .785714286         .6
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000         20 .928571429          1
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000         20 .928571429          1
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950         30 .142857143 .166666667
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250         30 .357142857         .5
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250         30 .357142857         .5

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL     DEPTNO         P1         P2
---------- ---------- --------- ---------- ------------------ ---------- ---------- ---------- ----------
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500         30         .5 .666666667
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600         30 .571428571 .833333333
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850         30 .714285714          1

14 rows selected.
```

4.12 PERCENT_RANK

1）语法结构

Analytic Syntax

***percent_rank_analytic*::=**



该函数和CUME_DIST函数类似，返回值的范围是0到1，使用该函数的任何集合的第一行是0。计算逻辑为：对于某一行R，总行数为N，值为（R-1）/（N-1）。

2）示例

```
SQL> SELECT empno,
       ename,
       job,
       mgr,
       hiredate,
       sal,
       deptno,
       percent_rank() over(ORDER BY sal) p1,
       percent_rank() over(PARTITION BY deptno ORDER BY sal) p2
  FROM emp
 ORDER BY deptno, sal;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	P1	P2
7934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300	10	.384615385	0
7782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450	10	.615384615	.5
7839	KING	PRESIDENT		1981-11-17 00:00:00	5000	10	1	1
7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800	20	0	0
7876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100	20	.153846154	.25
7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975	20	.769230769	.5
7902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000	20	.846153846	.75
7788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000	20	.846153846	.75
7900	JAMES	CLERK	7698	1981-12-03 00:00:00	950	30	.076923077	0
7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	30	.230769231	.2
7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	30	.230769231	.2

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO	P1	P2
7844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500	30	.461538462	.6
7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	30	.538461538	.8
7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850	30	.692307692	1

14 rows selected.