

Note: Small fonts in this presentation



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

1

ORACLE®



**Connor McDonald**  
@connor\_mc\_d



Partitioning an existing index

🕒 September 6, 2016

[connormcdonald.wordpress.com](http://connormcdonald.wordpress.com)

<https://www.youtube.com/c/ConnorMcDonaldOracle>

[Advanced Search](#)  
[Language Tools](#)**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

**Connor McDonald****Position: Second Row****D.O.B: 27/08/1988****Height: 6ft 4"****Weight: 18st 0lb**  
[Advanced Search](#)  
[Language Tools](#)**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# Connor McDonald



Position: Second Row

D.O.B: 2/7/94

Height: 6'1"

Weight: 125 lbs

## LEHIGH WRESTLING

TICKETS ALUMNI MERCHANDISE CAMPS AND CLINICS

Women's Basketball: 2010 Patriot League Champions!



### Connor McDonald

Year: Fr./Fr.  
Weight: 125  
Major: Arts and Sciences  
Hometown: Rehoboth, Del.  
Previous School: Sussex Central

In secondary school: Three-time Delaware state champion for Sussex Central... amassed over 150 career wins... won a junior national championship in 2006... ranked third in his weight class and 24th nationally by *InterMat*... won the Dave Schultz Award for the state of Delaware.

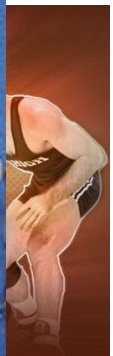


Google

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## IOWA STATE FAIR 2008



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Ask Tom Home x

https://asktom.oracle.com/pls/apex/f?p=100:1:0

Apps Quick 11g 11g doc 10g doc 11g2 doc 12.1 doc

**ORACLE Ask Tom** Sig

Questions Archives Popular Hot Resources About

search questions and answers Go Advanced Search Ask a Question

Recent Questions

Subject	Last Updated	First Asked
<a href="#">Correlated subquery</a>	107 minutes ago	3 hours ago
<a href="#">Materialized view containing LONG datatype,</a>	2 hours ago	3 hours ago
<a href="#">New Request feature in Oracle SQL</a>	4 hours ago	10 hours ago
<a href="#">Query related issue</a>	14 hours ago	21 hours ago
<a href="#">Regarding truncate operation</a>	17 hours ago	20 hours ago
<a href="#">External tables</a>	17 hours ago	23 hours ago
<a href="#">insert /*+append*/ into table ,can not insert null into table</a>	22 hours ago	26 hours ago
<a href="#">building tables dynamically for test data</a>	43 hours ago	2 days ago
<a href="#">GUID as the primary key in an active-active environment</a>	3 days ago	3 days ago
<a href="#">Purging non partitioned tables</a>	3 days ago	3 days ago


Monthly Recap

61 new questions

62 followups

19 followup responses

Answers Team >

 **Connor McDonald**  
@connor\_mc\_d

After starting as a COBOL programmer, Connor got hooked on Oracle in the 90's on Forms, but the database is his real passion. He's gone from Ace Director to SQL Advocate to Oracle.

# ANALYTIC SQL

*"SQL ... really ?"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |



TIBCO<sup>TM</sup> Spotfire<sup>®</sup>



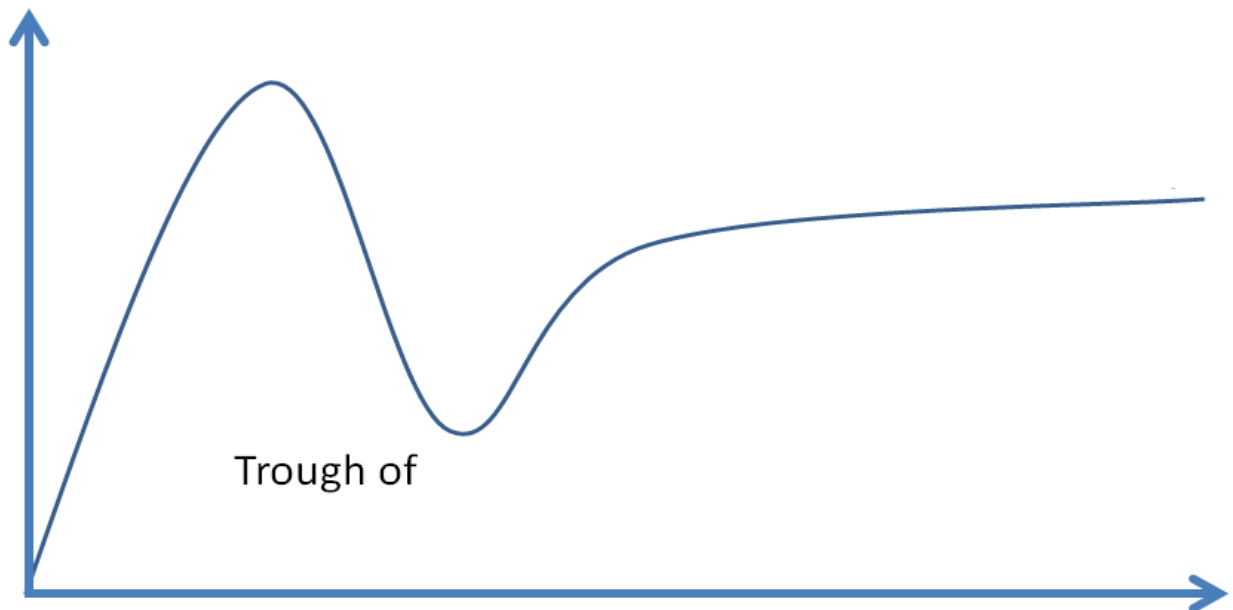
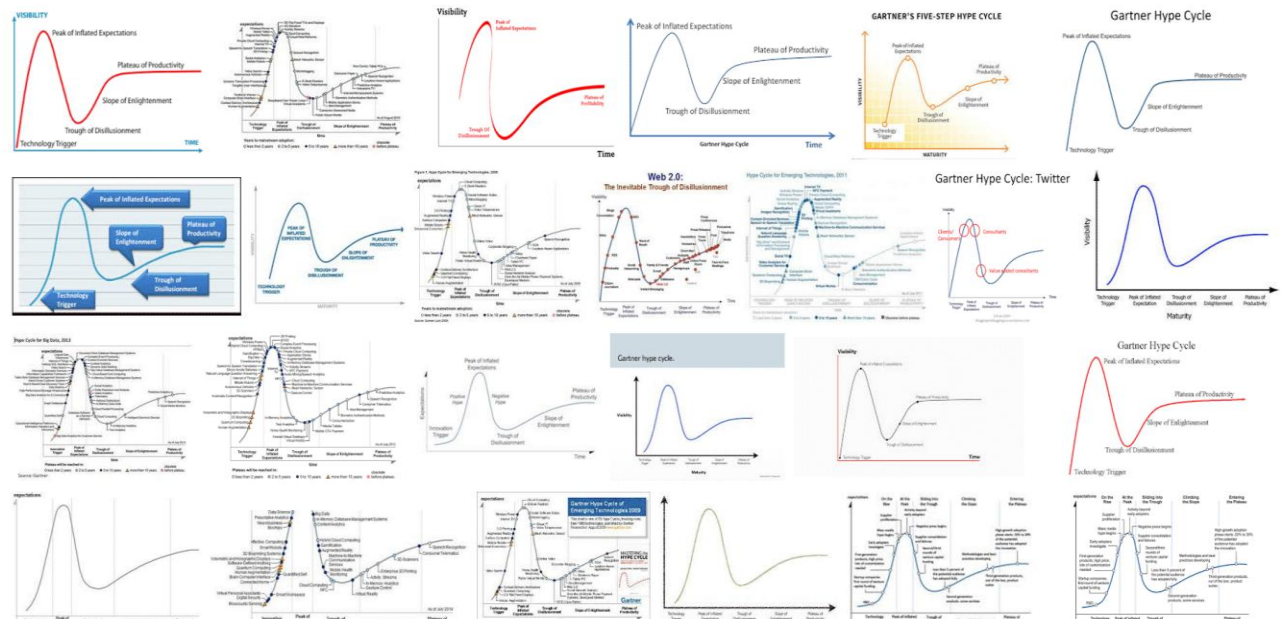
Qlik<sup>®</sup>



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |







sometimes...



The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, followed by a registered trademark symbol (®). The logo is centered within a red rectangular box, which is itself centered on a black background.

...get it wrong

A small version of the Oracle logo, featuring the word "ORACLE" in white, uppercase, sans-serif font, followed by a registered trademark symbol (®), all contained within a red rectangular box.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# Data warehousing guide

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Contents - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/oracle/doc/11.2/server.112/e10810/toc.htm

Contents

Oracle® Database Data Warehousing Guide  
11g Release 2 (11.2)  
Part Number E10810-02

Home Book List Index Master Index Contents

Next

## Contents

### [Title and Copyright Information](#)

### [Preface](#)

[Audience](#)  
[Documentation Accessibility](#)  
[Related Documents](#)  
[Conventions](#)

### [What's New in Oracle Database?](#)

[Oracle Database 11g Release 2 \(11.2\) New Features in Data Warehousing](#)  
[Oracle Database 11g Release 1 \(11.1\) New Features in Data Warehousing](#)

## [21. SQL for Analysis and Reporting](#)

[Overview of SQL for Analysis and Reporting](#)  
[Ranking, Windowing, and Reporting Functions](#)  
[Ranking](#)  
[RANK and DENSE\\_RANK Functions](#)  
[Bottom N Ranking](#)  
[CUME\\_DIST Function](#)  
[PERCENT\\_RANK Function](#)  
[NTILE Function](#)  
[Reporting Functions](#)

## simple syntax



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
<function> ( <arg>,<arg>,... )  
OVER (  
    <partition clause>  
    <sorting clause>  
    <windowing clause>  
)
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

17

That's it !



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

18

quick example #1



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

19

# employees by salary

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

20

```
SQL> select empno, ename, job, hiredate, sal
2  from emp
3  order by sal;
```

EMPNO	ENAME	JOB	HIREDATE	SAL
7369	SMITH	CLERK	17-DEC-80	800
7900	JAMES	CLERK	03-DEC-81	950
7876	ADAMS	CLERK	12-JAN-83	1100
7521	WARD	SALESMAN	22-FEB-81	1250
7654	MARTIN	SALESMAN	28-SEP-81	1250
7934	MILLER	CLERK	23-JAN-82	1300
7844	TURNER	SALESMAN	08-SEP-81	1500
7499	ALLEN	SALESMAN	20-FEB-81	1600
7782	CLARK	MANAGER	09-JUN-81	2450
7698	BLAKE	MANAGER	01-MAY-81	2850
7566	JONES	MANAGER	02-APR-81	2975
7902	FORD	ANALYST	03-DEC-81	3000
7788	SCOTT	ANALYST	09-DEC-82	3000
7839	KING	PRESIDENT	17-NOV-81	5000

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

*"give me the hiring sequence"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

22

EMPNO	ENAME	JOB	HIREDATE	SAL	HIRE_SEQ
7369	SMITH	CLERK	17-DEC-80	800	1
7900	JAMES	CLERK	03-DEC-81	950	10
7876	ADAMS	CLERK	12-JAN-83	1100	14
7521	WARD	SALESMAN	22-FEB-81	1250	3
7654	MARTIN	SALESMAN	28-SEP-81	1250	8
7934	MILLER	CLERK	23-JAN-82	1300	12
7844	TURNER	SALESMAN	08-SEP-81	1500	7
7499	ALLEN	SALESMAN	20-FEB-81	1600	2
7782	CLARK	MANAGER	09-JUN-81	2450	6
7698	BLAKE	MANAGER	01-MAY-81	2850	5
7566	JONES	MANAGER	02-APR-81	2975	4
7902	FORD	ANALYST	03-DEC-81	3000	10
7788	SCOTT	ANALYST	09-DEC-82	3000	13
7839	KING	PRESIDENT	17-NOV-81	5000	9

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

SMITH was hired "first"

EMPNO	ENAME	JOB	HIREDATE	SAL	HIRE_SEQ
7369	SMITH	CLERK	17-DEC-80	800	1
7900	JAMES	CLERK	03-DEC-81	950	10
7876	ADAMS	CLERK	12-JAN-83	1100	14
7521	WARD	SALESMAN	22-FEB-81	1250	3
7654	MARTIN	SALESMAN	28-SEP-81	1250	8
7934	MILLER	CLERK	23-JAN-82	1300	12
7844	TURNER	SALESMAN	08-SEP-81	1500	7
7499	ALLEN	SALESMAN	20-FEB-81	1600	2
7782	CLARK	MANAGER	09-JUN-81	2450	6
7698	BLAKE	MANAGER	01-MAY-81	2850	5
7566	JONES	MANAGER	02-APR-81	2975	4
7902	FORD	ANALYST	03-DEC-81	3000	10
7788	SCOTT	ANALYST	09-DEC-82	3000	13
7839	KING	PRESIDENT	17-NOV-81	5000	9

ADAMS was hired "last"

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

non-trivial

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

24



```

SQL> select e.empno, e.ename, e.job,
2         e.hiredate, e.sal, x.seq
3   from emp e,
4        ( select e2.empno, count(*) seq
5          from   emp e1, emp e2
6          where  e1.hiredate <= e2.hiredate
7          group by e2.empno
8        ) x
9   where e.empno = x.empno
10  order by sal;

```

huh ?



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Id	Operation	Name	Rows	Bytes
0	SELECT STATEMENT		10	1390
1	<b>SORT GROUP BY</b>		10	1390
* 2	<b>HASH JOIN</b>		10	1390
3	<b>MERGE JOIN</b>		10	310
4	<b>SORT JOIN</b>		14	126
5	TABLE ACCESS FULL	EMP	14	126
* 6	<b>SORT JOIN</b>		14	308
7	TABLE ACCESS FULL	EMP	14	308
8	TABLE ACCESS FULL	EMP	14	1512

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select empno, ename, job, hiredate, sal,
3  from emp
4  order by sal;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	HIRE_SEQ
7369	SMITH	CLERK	17-DEC-80	800	1
7900	JAMES	CLERK	03-DEC-81	950	10
7876	ADAMS	CLERK	12-JAN-83	1100	14
7521	WARD	SALESMAN	22-FEB-81	1250	3
7654	MARTIN	SALESMAN	28-SEP-81	1250	8
7934	MILLER	CLERK	23-JAN-82	1300	12
7844	TURNER	SALESMAN	08-SEP-81	1500	7
7499	ALLEN	SALESMAN	20-FEB-81	1600	2
7782	CLARK	MANAGER	09-JUN-81	2450	6
7698	BLAKE	MANAGER	01-MAY-81	2850	5
7566	JONES	MANAGER	02-APR-81	2975	4
7902	FORD	ANALYST	03-DEC-81	3000	10
7788	SCOTT	ANALYST	09-DEC-82	3000	13
7839	KING	PRESIDENT	17-NOV-81	5000	9

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select empno, ename, job, hiredate, sal,
2  rank() OVER (order by hiredate) as hire_seq
3  from emp
4  order by sal;
```

EMPNO	ENAME	JOB	HIREDATE	SAL	HIRE_SEQ
7369	SMITH	CLERK	17-DEC-80	800	1
7900	JAMES	CLERK	03-DEC-81	950	10
7876	ADAMS	CLERK	12-JAN-83	1100	14
7521	WARD	SALESMAN	22-FEB-81	1250	3
7654	MARTIN	SALESMAN	28-SEP-81	1250	8
7934	MILLER	CLERK	23-JAN-82	1300	12
7844	TURNER	SALESMAN	08-SEP-81	1500	7
7499	ALLEN	SALESMAN	20-FEB-81	1600	2
7782	CLARK	MANAGER	09-JUN-81	2450	6
7698	BLAKE	MANAGER	01-MAY-81	2850	5
7566	JONES	MANAGER	02-APR-81	2975	4
7902	FORD	ANALYST	03-DEC-81	3000	10
7788	SCOTT	ANALYST	09-DEC-82	3000	13
7839	KING	PRESIDENT	17-NOV-81	5000	9

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

function

```
rank() OVER (  
  order by hire_date) as hire_seq
```

sorting clause

## Functions *for ranking*

RANK

1, 2, 3, 3, 5, ...

## Functions *for ranking*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

30

RANK

1, 2, 3, 3, 5, ...

DENSE\_RANK

1, 2, 3, 3, 4, ...

## Functions *for ranking*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

31

RANK

DENSE\_RANK

CUME\_DIST

```
SQL> select ename, sal,
2      100*cume_dist() over ( order by sal ) as pct
3  from emp
4  order by sal;
```

ENAME	SAL	PCT
SMITH	800	7.14
JAMES	950	14.29
ADAMS	1100	21.43
WARD	1250	35.71
MARTIN	1250	35.71
MILLER	1300	42.86
TURNER	1500	50.00
ALLEN	1600	57.14
CLARK	2450	64.29
BLAKE	2850	71.43
JONES	2975	78.57
FORD	3000	92.86
SCOTT	3000	92.86
KING	5000	100.00

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

RANK

DENSE\_RANK

CUME\_DIST

PERCENT\_RANK

```
SQL> select ename, sal,
2      100*percent_rank() over ( order by sal ) pct
3  from emp
4  order by ename;
```

ENAME	SAL	PCT
ADAMS	1100	15.38
ALLEN	1600	53.85
BLAKE	2850	69.23
CLARK	2450	61.54
FORD	3000	84.62
JAMES	950	7.69
JONES	2975	76.92
KING	5000	100.00
MARTIN	1250	23.08
MILLER	1300	38.46
SCOTT	3000	84.62
SMITH	800	.00
TURNER	1500	46.15
WARD	1250	23.08

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

RANK

DENSE\_RANK

CUME\_DIST

PERCENT\_RANK

NTILE

```
SQL> select ename, sal,
2      ntile(4) over ( order by sal ) as quartile
3  from emp
4  order by ename;
```

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

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

RANK 1, 2, 3, 3, 5, ...

DENSE\_RANK 1, 2, 3, 3, 4, ...

CUME\_DIST

PERCENT\_RANK

NTILE

ROW\_NUMBER 1, 2, 3, 4, 5, ...

*Functions for ranking*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## Functions *for aggregation*

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

36

SUM  
AVERAGE  
MIN  
MAX  
COUNT

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

37



## quick example #2

*"department salaries,  
running total  
by employee name"*

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5  from emp
6  order by deptno, ename;

```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5  from emp
6  order by deptno, ename;

```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5 from emp
6 order by deptno, ename;

```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
-----					
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5 from emp
6 order by deptno, ename;

```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
-----					
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5  from emp
6  order by deptno, ename;
```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

function

partition clause

```
sum(sal) OVER (
  partition by deptno
  order by ename) as running_total
```

sorting clause

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

45



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

46

```
SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5  from emp
6  order by deptno, ename;
```

DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400


ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3      partition by deptno
4      order by ename) as running_total
5 from emp
6 order by deptno, hiredate;

```



DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7369	SMITH	CLERK	800	10875
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
30	7499	ALLEN	SALESMAN	1600	1600
30	7521	WARD	SALESMAN	1250	9400
30	7654	MARTIN	SALESMAN	1250	6650
30	7698	BLAKE	MANAGER	2850	4450
30	7844	TURNER	SALESMAN	1500	8150
30	7900	JAMES	CLERK	950	5400


ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3      partition by deptno
4      order by ename) as running_total
5 from emp
6 order by deptno, hiredate;

```



DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7369	SMITH	CLERK	800	10875
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
30	7499	ALLEN	SALESMAN	1600	1600
30	7521	WARD	SALESMAN	1250	9400
30	7654	MARTIN	SALESMAN	1250	6650
30	7698	BLAKE	MANAGER	2850	4450
30	7844	TURNER	SALESMAN	1500	8150
30	7900	JAMES	CLERK	950	5400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# lots of power

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

49

```
SQL> select deptno, job, ename, sal,
2         sum(sal) over () total_sal,
3         sum(sal) over
4           ( partition by deptno) sal_by_dept,
5         sum(sal) over
6           ( partition by deptno, job) sal_by_dept_job
7   from emp
8  order by deptno, job;
```

DEPTNO	JOB	ENAME	SAL	TOTAL_SAL	SAL_BY_DEPT	SAL_BY_DEPT_JOB
10	CLERK	MILLER	1300	29025	8750	1300
10	MANAGER	CLARK	2450	29025	8750	2450
10	PRESIDENT	KING	5000	29025	8750	5000
20	ANALYST	SCOTT	3000	29025	10875	6000
20	ANALYST	FORD	3000	29025	10875	6000
20	CLERK	ADAMS	1100	29025	10875	1900
20	CLERK	SMITH	800	29025	10875	1900
20	MANAGER	JONES	2975	29025	10875	2975
30	CLERK	JAMES	950	29025	9400	950
30	MANAGER	BLAKE	2850	29025	9400	2850
30	SALESMAN	TURNER	1500	29025	9400	5600
30	SALESMAN	MARTIN	1250	29025	9400	5600
30	SALESMAN	WARD	1250	29025	9400	5600
30	SALESMAN	ALLEN	1600	29025	9400	5600

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |



and a whole lot more...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

STATS\_T\_TEST\_...      STDDEV\_SAMP      COVAR\_SAMP      VAR\_POP

GROUP\_ID      MEDIAN      STATS\_MODE

STDDEV

STATS\_CROSSTAB      CORR      STATS\_BINOMIAL\_TEST      GROUPING\_ID      REGR\_ ...

COLLECT

STATS\_MW\_TEST

STATS\_WSR\_TEST      PERCENTILE\_CONT      STDDEV\_POP

COVAR\_POP      VARIANCE      GROUPING      STATS\_ONE\_WAY\_ANOVA

STATS\_KS\_TEST      PERCENTILE\_DISC      STATS\_F\_TEST      VAR\_SAMP

and a whole lot more...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

there is a cost

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select deptno, empno, ename, job, sal,  
2      sum(sal) OVER ( ... ),  
3      max(hiredate) over ( ... ),  
4      stddev(commission) over ( .... )  
      ...  
      ...  
      ...  
25 from emp
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

a **lot** of work...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

and for the DBA's...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select deptno, empno, ename, job, sal
2   from emp
3   order by deptno, empno;
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		12	1044	4 (25)
1	<b>SORT ORDER BY</b>		12	1044	4 (25)
2	TABLE ACCESS FULL	EMP	12	1044	3 (0)

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select deptno, empno, ename, job, sal,
2   sum(sal) OVER (
3     partition by deptno
4     order by ename) as running_total
5   from emp
6   order by deptno, empno
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		12	468	3 (0)
1	<b>WINDOW SORT</b>		12	468	3 (0)
2	TABLE ACCESS FULL	EMP	12	468	3 (0)

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## some possible sorting anomalies

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## MOS

workarea\_size\_policy  
\_smm\_isort\_cap  
\_smm\_max\_size  
\_newsort\_enabled  
\_smm\_auto\_min\_io\_size  
\_smm\_auto\_max\_io\_size

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

important note

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

aggregations

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

calculated

not

restrictive

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

62

conventional aggregation

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

63



```
SQL> select deptno, sum(sal)
2   from   emp
3   group  by deptno;
```

DEPTNO	SUM(SAL)
10	8750
20	10875
30	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

64

```
SQL> select ename, deptno,
2       sum(sal) over
3       ( partition by deptno) as deptsal
4   from emp
5   order by deptno;
```

ENAME	DEPTNO	DEPTSAL
CLARK	10	8750
KING	10	8750
MILLER	10	8750
JONES	20	10875
FORD	20	10875
ADAMS	20	10875
SMITH	20	10875
SCOTT	20	10875
WARD	30	9400
TURNER	30	9400
ALLEN	30	9400
JAMES	30	9400
BLAKE	30	9400
MARTIN	30	9400

Still 14 rows !



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

65

two kinds

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

66

aggregation

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

67

reporting aggregation

"same aggregate for each row in a partition"




reporting aggregation

```
SQL> select ename, deptno,  
2      sum(sal) over ( partition by deptno) as deptsal  
3 from emp  
4 order by deptno;
```

ENAME	DEPTNO	DEPTSAL
CLARK	10	8750
KING	10	8750
MILLER	10	8750
JONES	20	10875
FORD	20	10875
ADAMS	20	10875
SMITH	20	10875
SCOTT	20	10875
WARD	30	9400
TURNER	30	9400
ALLEN	30	9400
JAMES	30	9400
BLAKE	30	9400
MARTIN	30	9400

same for  
each row  
in partition

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

69

## windowing aggregation

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

70

"changing aggregate for each row in a partition"

**windowing** aggregation

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

70

```
<function> ( <arg>,<arg>,... )  
OVER (  
    <partition clause>  
    <sorting clause>  
    <windowing clause>  
)
```

defines how "broadly" the  
aggregating function applies

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## *"salary cumulative total"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select
  2   empno, ename, sal,
  3   sum(sal)
  4   over ( order by empno
  5         rows between unbounded preceding and current row ) as cumtot
  6 from   emp
  7 order by empno;
```

EMPNO	ENAME	SAL	CUMTOT
7369	SMITH	800	800
7499	ALLEN	1600	2400
7521	WARD	1250	3650
7566	JONES	2975	6625
7654	MARTIN	1250	7875
7698	BLAKE	2850	10725
7782	CLARK	2450	13175
7788	SCOTT	3000	16175
7839	KING	5000	21175
7844	TURNER	1500	22675
7876	ADAMS	1100	23775
7900	JAMES	950	24725
7902	FORD	3000	27725
7934	MILLER	1300	29025

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

73

```

SQL> select
  2   empno, ename, sal,
  3   sum(sal)
  4   over ( order by empno
  5         rows between unbounded preceding and current row ) as cumtot
  6 from   emp
  7 order by empno;

```

EMPNO	ENAME	SAL	CUMTOT
7369	SMITH	800	800
7499	ALLEN	1600	2400
7521	WARD	1250	3650
7566	JONES	2975	6625
7654	MARTIN	1250	7875
7698	BLAKE	2850	10725
7782	CLARK	2450	13175
7788	SCOTT	3000	16175
7839	KING	5000	21175
7844	TURNER	1500	22675
7876	ADAMS	1100	23775
7900	JAMES	950	24725
7902	FORD	3000	27725
7934	MILLER	1300	29025

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

74

```

SQL> select
  2   empno, ename, sal,
  3   sum(sal)
  4   over ( order by empno
  5         rows between unbounded preceding and current row ) as cumtot
  6 from   emp
  7 order by empno;

```

EMPNO	ENAME	SAL	CUMTOT
7369	SMITH	800	800
7499	ALLEN	1600	2400
7521	WARD	1250	3650
7566	JONES	2975	6625
7654	MARTIN	1250	7875
7698	BLAKE	2850	10725
7782	CLARK	2450	13175
7788	SCOTT	3000	16175
7839	KING	5000	21175
7844	TURNER	1500	22675
7876	ADAMS	1100	23775
7900	JAMES	950	24725
7902	FORD	3000	27725
7934	MILLER	1300	29025

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

75

```

SQL> select
2   empno, ename, sal,
3   sum(sal)
4   over ( order by empno
5         rows between unbounded preceding and current row ) as cumtot
6 from   emp
7 order by empno;

```

EMPNO	ENAME	SAL	CUMTOT
7369	SMITH	800	800
7499	ALLEN	1600	2400
7521	WARD	1250	3650
7566	JONES	2975	6625
7654	MARTIN	1250	7875
7698	BLAKE	2850	10725
7782	CLARK	2450	13175
7788	SCOTT	3000	16175
7839	KING	5000	21175
7844	TURNER	1500	22675
7876	ADAMS	1100	23775
7900	JAMES	950	24725
7902	FORD	3000	27725
7934	MILLER	1300	29025

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

76

*"sum across 3 rows"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

77



```

SQL> select deptno, ename, hiredate, sal,
2      sum(sal) over (
3        partition by deptno
4        order by hiredate
5        rows between 1 preceding and 1 following) as x
6      from emp
7      order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	7450
10	KING	17-NOV-81	5000	8750
10	MILLER	23-JAN-82	1300	6300
20	SMITH	17-DEC-80	800	3775
20	JONES	02-APR-81	2975	6775
20	FORD	03-DEC-81	3000	8975
20	SCOTT	09-DEC-82	3000	7100
20	ADAMS	12-JAN-83	1100	4100
30	ALLEN	20-FEB-81	1600	2850
30	WARD	22-FEB-81	1250	5700
30	BLAKE	01-MAY-81	2850	5600
30	TURNER	08-SEP-81	1500	5600
30	MARTIN	28-SEP-81	1250	3700
30	JAMES	03-DEC-81	950	2200

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2      sum(sal) over (
3        partition by deptno
4        order by hiredate
5        rows between 1 preceding and 1 following) as x
6      from emp
7      order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	7450
10	KING	17-NOV-81	5000	8750
10	MILLER	23-JAN-82	1300	6300
20	SMITH	17-DEC-80	800	3775
20	JONES	02-APR-81	2975	6775
20	FORD	03-DEC-81	3000	8975
20	SCOTT	09-DEC-82	3000	7100
20	ADAMS	12-JAN-83	1100	4100
30	ALLEN	20-FEB-81	1600	2850
30	WARD	22-FEB-81	1250	5700
30	BLAKE	01-MAY-81	2850	5600
30	TURNER	08-SEP-81	1500	5600
30	MARTIN	28-SEP-81	1250	3700
30	JAMES	03-DEC-81	950	2200

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2      sum(sal) over (
3        partition by deptno
4        order by hiredate
5        rows between 1 preceding and 1 following) as x
6      from emp
7      order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	7450
10	KING	17-NOV-81	5000	8750
10	MILLER	23-JAN-82	1300	6300
20	SMITH	17-DEC-80	800	3775
20	JONES	02-APR-81	2975	6775
20	FORD	03-DEC-81	3000	8975
20	SCOTT	09-DEC-82	3000	7100
20	ADAMS	12-JAN-83	1100	4100
30	ALLEN	20-FEB-81	1600	2850
30	WARD	22-FEB-81	1250	5700
30	BLAKE	01-MAY-81	2850	5600
30	TURNER	08-SEP-81	1500	5600
30	MARTIN	28-SEP-81	1250	3700
30	JAMES	03-DEC-81	950	2200

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2      sum(sal) over (
3        partition by deptno
4        order by hiredate
5        rows between 1 preceding and 1 following) as x
6      from emp
7      order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	7450
10	KING	17-NOV-81	5000	8750
10	MILLER	23-JAN-82	1300	6300
20	SMITH	17-DEC-80	800	3775
20	JONES	02-APR-81	2975	6775
20	FORD	03-DEC-81	3000	8975
20	SCOTT	09-DEC-82	3000	7100
20	ADAMS	12-JAN-83	1100	4100
30	ALLEN	20-FEB-81	1600	2850
30	WARD	22-FEB-81	1250	5700
30	BLAKE	01-MAY-81	2850	5600
30	TURNER	08-SEP-81	1500	5600
30	MARTIN	28-SEP-81	1250	3700
30	JAMES	03-DEC-81	950	2200

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

## "6 month moving average"

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

82

```
SQL> select deptno, ename, hiredate, sal,
2  avg(sal) over (
3    partition by deptno
4    order by hiredate
5    range between interval '6' month preceding and current row ) x
6  from emp
7  order by deptno, hiredate;
```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	2450
10	KING	17-NOV-81	5000	3725
10	MILLER	23-JAN-82	1300	3150
20	SMITH	17-DEC-80	800	800
20	JONES	02-APR-81	2975	1888
20	FORD	03-DEC-81	3000	3000
20	SCOTT	09-DEC-82	3000	3000
20	ADAMS	12-JAN-83	1100	2050
30	ALLEN	20-FEB-81	1600	1600
30	WARD	22-FEB-81	1250	1425
30	BLAKE	01-MAY-81	2850	1900
30	TURNER	08-SEP-81	1500	2175
30	MARTIN	28-SEP-81	1250	1867
30	JAMES	03-DEC-81	950	1233

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2   avg(sal) over (
3     partition by deptno
4     order by hiredate
5     range between interval '6' month preceding and current row ) x
6   from emp
7   order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	2450
10	KING	17-NOV-81	5000	3725
10	MILLER	23-JAN-82	1300	3150
20	SMITH	17-DEC-80	800	800
20	JONES	02-APR-81	2975	1888
20	FORD	03-DEC-81	3000	3000
20	SCOTT	09-DEC-82	3000	3000
20	ADAMS	12-JAN-83	1100	2050
30	ALLEN	20-FEB-81	1600	1600
30	WARD	22-FEB-81	1250	1425
30	BLAKE	01-MAY-81	2850	1900
30	TURNER	08-SEP-81	1500	2175
30	MARTIN	28-SEP-81	1250	1867
30	JAMES	03-DEC-81	950	1233

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2   avg(sal) over (
3     partition by deptno
4     order by hiredate
5     range between interval '6' month preceding and current row ) x
6   from emp
7   order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	2450
10	KING	17-NOV-81	5000	3725
10	MILLER	23-JAN-82	1300	3150
20	SMITH	17-DEC-80	800	800
20	JONES	02-APR-81	2975	1888
20	FORD	03-DEC-81	3000	3000
20	SCOTT	09-DEC-82	3000	3000
20	ADAMS	12-JAN-83	1100	2050
30	ALLEN	20-FEB-81	1600	1600
30	WARD	22-FEB-81	1250	1425
30	BLAKE	01-MAY-81	2850	1900
30	TURNER	08-SEP-81	1500	2175
30	MARTIN	28-SEP-81	1250	1867
30	JAMES	03-DEC-81	950	1233

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, hiredate, sal,
2      avg(sal) over (
3        partition by deptno
4          order by hiredate
5          range between interval '6' month preceding and current row ) x
6      from emp
7      order by deptno, hiredate;

```

DEPTNO	ENAME	HIREDATE	SAL	X
10	CLARK	09-JUN-81	2450	2450
10	KING	17-NOV-81	5000	3725
10	MILLER	23-JAN-82	1300	3150
20	SMITH	17-DEC-80	800	800
20	JONES	02-APR-81	2975	1888
20	FORD	03-DEC-81	3000	3000
20	SCOTT	09-DEC-82	3000	3000
20	ADAMS	12-JAN-83	1100	2050
30	ALLEN	20-FEB-81	1600	1600
30	WARD	22-FEB-81	1250	1425
30	BLAKE	01-MAY-81	2850	1900
30	TURNER	08-SEP-81	1500	2175
30	MARTIN	28-SEP-81	1250	1867
30	JAMES	03-DEC-81	950	1233

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# dynamic windows

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

87

*"sum sales from previous  
(business) close-off day"*

```
SQL> create or replace
 2 function LAST_CLOSE(p_purchase_date date)
 3 return number is
 4 begin
 5     return
 6         case to_char(p_purchase_date, 'DY')
 7             when 'SUN' then 2
 8             when 'MON' then 3
 9             else 1
10         end;
11 end;
12 /
```

Function created.

```
SQL> select
  2   prod_id, cust_id,
  3   sum(amount_sold)
  4   over ( order by purchase_date
  5         range between LAST_CLOSE(purchase_date) preceding) as bus_tot
  6 from   sales
  7 /
```

## window boundaries

first\_value / last\_value

*"compare each salary with lowest  
across entire organisation  
and with in each department"*



```

SQL> select deptno, empno, ename, sal,
2      first_value(sal) over ( order by sal
3      range unbounded preceding ) lo_sal,
4      first_value(sal) over ( partition by deptno
5      order by sal
6      range unbounded preceding) lo_dept_sal
7 from emp
8 order by deptno, sal;

```

DEPTNO	EMPNO	ENAME	SAL	LO_SAL	LO_DEPT_SAL
10	7782	CLARK	2450	800	2450
10	7839	KING	5000	800	2450
20	7369	SMITH	800	800	800
20	7876	ADAMS	1100	800	800
20	7566	JONES	2975	800	800
20	7902	FORD	3000	800	800
20	7788	SCOTT	3000	800	800
30	7900	JAMES	950	800	950
30	7521	WARD	1250	800	950
30	7654	MARTIN	1250	800	950
30	7844	TURNER	1500	800	950
30	7499	ALLEN	1600	800	950
30	7698	BLAKE	2850	800	950
40	7934	MILLER	1300	800	1300

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, sal,
2      100 * sal / first_value(sal) over ( order by sal
3      range unbounded preceding ) lo_sal,
4      100 * sal / first_value(sal) over ( partition by deptno
5      order by sal
6      range unbounded preceding) lo_dept_sal
7 from emp
8 order by deptno, sal;

```

DEPTNO	EMPNO	ENAME	SAL	LO_SAL_PCT	LO_DEPT_SAL_PCT
10	7782	CLARK	2450	306.25	100.00
10	7839	KING	5000	625.00	204.08
20	7369	SMITH	800	100.00	100.00
20	7876	ADAMS	1100	137.50	137.50
20	7566	JONES	2975	371.88	371.88
20	7902	FORD	3000	375.00	375.00
20	7788	SCOTT	3000	375.00	375.00
30	7900	JAMES	950	118.75	100.00
30	7521	WARD	1250	156.25	131.58
30	7654	MARTIN	1250	156.25	131.58
30	7844	TURNER	1500	187.50	157.89
30	7499	ALLEN	1600	200.00	168.42
30	7698	BLAKE	2850	356.25	300.00
40	7934	MILLER	1300	162.50	100.00

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

11.2+

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

nth\_value

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, empno, ename, sal,
2      100 * sal / nth_value(sal,2)
3          over ( order by sal ) lo_sal_pct,
4      100 * sal / nth_value(sal,3) from last
5          over ( partition by deptno
6                order by sal ) hi_dept_sal_pct
7 from    emp
8 order by deptno, sal;

```

DEPTNO	EMPNO	ENAME	SAL	LO_SAL_PCT	HI_DEPT_SAL_PCT
10	7934	MILLER	1300	136.84	
10	7782	CLARK	2450	257.89	188.46
10	7839	KING	5000	526.32	204.08
20	7369	SMITH	800		
20	7876	ADAMS	1100	115.79	137.50
20	7566	JONES	2975	313.16	270.45
20	7902	FORD	3000	315.79	100.00
20	7788	SCOTT	3000	315.79	100.00
30	7900	JAMES	950	100.00	
30	7521	WARD	1250	131.58	100.00
30	7654	MARTIN	1250	131.58	100.00
30	7844	TURNER	1500	157.89	120.00
30	7499	ALLEN	1600	168.42	106.67
30	7698	BLAKE	2850	300.00	178.13

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# ignore nulls extension

ORACLE

99 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

99

```
SQL> select empno, ename, sal, deptno
2   from emp
3   order by sal;
```

EMPNO	ENAME	SAL	DEPTNO
7369	SMITH	800	10
7900	JAMES	950	
7876	ADAMS	1100	
7521	WARD	1250	20
7654	MARTIN	1250	
7934	MILLER	1300	
7844	TURNER	1500	
7499	ALLEN	1600	30
7782	CLARK	2450	
7698	BLAKE	2850	
7566	JONES	2975	
7788	SCOTT	3000	
7902	FORD	3000	
7839	KING	5000	40



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

100

```
SQL> select empno, ename, sal, deptno,
2   last_value(deptno IGNORE NULLS)
3   over (order by sal) as last_dept
4   from emp
5   order by sal
```

EMPNO	ENAME	SAL	DEPTNO	LAST_DEPT
7369	SMITH	800	10	10
7900	JAMES	950		10
7876	ADAMS	1100		10
7521	WARD	1250	20	20
7654	MARTIN	1250		20
7934	MILLER	1300		20
7844	TURNER	1500		20
7499	ALLEN	1600	30	30
7782	CLARK	2450		30
7698	BLAKE	2850		30
7566	JONES	2975		30
7788	SCOTT	3000		30
7902	FORD	3000		30
7839	KING	5000	40	40

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select last_dept, count(*)
2   from
3   ( select
4       last_value(deptno ignore nulls)
5       over (order by sal) as last_dept
6   from emp2
7   )
8   group by last_dept;
```

LAST_DEPT	COUNT(*)
30	6
20	4
40	1
10	3



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

102

# implicit windows



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# recall

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

104

```
SQL> select ename, deptno,  
2      sum(sal) over ( partition by deptno ) as deptsal  
3  from emp  
4  order by deptno;
```

ENAME	DEPTNO	DEPTSAL
-----	-----	-----
CLARK	10	8750
KING	10	8750
MILLER	10	8750
JONES	20	10875
FORD	20	10875
ADAMS	20	10875
SMITH	20	10875
SCOTT	20	10875
WARD	30	9400
TURNER	30	9400
ALLEN	30	9400
JAMES	30	9400
BLAKE	30	9400
MARTIN	30	9400

reporting  
aggregate

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

105

```
SQL> select deptno, empno, ename, job, sal,
2      sum(sal) OVER (
3        partition by deptno
4        order by ename) as running_total
5      from emp
```

windowing  
aggregate

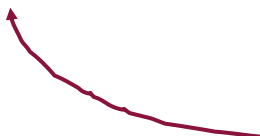


DEPTNO	EMPNO	ENAME	JOB	SAL	RUNNING_TOTAL
10	7782	CLARK	MANAGER	2450	2450
10	7839	KING	PRESIDENT	5000	7450
10	7934	MILLER	CLERK	1300	8750
20	7876	ADAMS	CLERK	1100	1100
20	7902	FORD	ANALYST	3000	4100
20	7566	JONES	MANAGER	2975	7075
20	7788	SCOTT	ANALYST	3000	10075
20	7369	SMITH	CLERK	800	10875
30	7499	ALLEN	SALESMAN	1600	1600
30	7698	BLAKE	MANAGER	2850	4450
30	7900	JAMES	CLERK	950	5400
30	7654	MARTIN	SALESMAN	1250	6650
30	7844	TURNER	SALESMAN	1500	8150
30	7521	WARD	SALESMAN	1250	9400

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
<function> ( <arg>,<arg>,... )
OVER (
  <partition clause>
  <sorting clause>
  <windowing clause>
)
```




but I didn't specify  
one of these ?

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

IF this is an  
aggregate function ...




```
<function> ( <arg>,<arg>,... )  
OVER (  
  <partition clause>  
  <sorting clause>  
  <windowing clause>  
)
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

IF this is an  
aggregate function ...



```
<function> ( <arg>,<arg>,... )  
OVER (  
  <partition clause>  
  <sorting clause>  
  <windowing clause>  
)
```

AND you have included  
an ORDER BY clause ...



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |



IF this is an  
aggregate function ...

AND you have included  
an ORDER BY clause ...

```
<function> ( <arg>,<arg>,... )
OVER (
  <partition clause>
  <sorting clause>
  <windowing clause>
)
```

THEN you get one of  
these automatically !

range between unbounded preceding and current row

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

"OVER"

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

109



lag / lead

```
SQL> select
  2   empno, ename, hiredate, sal,
  3   lag(sal,1)
  4   over ( order by hiredate ) prev_hiree_sal
  5 from   emp
  6 order by hiredate;
```

EMPNO	ENAME	HIREDATE	SAL	PREV_HIREE_SAL
7369	SMITH	17-DEC-80	800	
7499	ALLEN	20-FEB-81	1600	800
7521	WARD	22-FEB-81	1250	1600
7566	JONES	02-APR-81	2975	1250
7698	BLAKE	01-MAY-81	2850	2975
7782	CLARK	09-JUN-81	2450	2850
7844	TURNER	08-SEP-81	1500	2450
7654	MARTIN	28-SEP-81	1250	1500
7839	KING	17-NOV-81	5000	1250
7900	JAMES	03-DEC-81	950	5000
7902	FORD	03-DEC-81	3000	950
7934	MILLER	23-JAN-82	1300	3000
7788	SCOTT	09-DEC-82	3000	1300
7876	ADAMS	12-JAN-83	1100	3000

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select
  2   empno, ename, hiredate, sal,
  3   lag(sal,1)
  4   over ( order by hiredate ) prev_hiree_sal
  5 from   emp
  6 order by hiredate;
```

EMPNO	ENAME	HIREDATE	SAL	PREV_HIREE_SAL
7369	SMITH	17-DEC-80	800	
7499	ALLEN	20-FEB-81	1600	800
7521	WARD	22-FEB-81	1250	1600
7566	JONES	02-APR-81	2975	1250
7698	BLAKE	01-MAY-81	2850	2975
7782	CLARK	09-JUN-81	2450	2850
7844	TURNER	08-SEP-81	1500	2450
7654	MARTIN	28-SEP-81	1250	1500
7839	KING	17-NOV-81	5000	1250
7900	JAMES	03-DEC-81	950	5000
7902	FORD	03-DEC-81	3000	950
7934	MILLER	23-JAN-82	1300	3000
7788	SCOTT	09-DEC-82	3000	1300
7876	ADAMS	12-JAN-83	1100	3000

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

11.2+

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

113

ignore nulls

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

114

pre 11.2

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

115

```
SQL> select deptno, ename, sal
```

```
2  from emp
```

```
3  order by deptno, empno;
```

DEPTNO	ENAME	SAL
10	CLARK	
10	KING	5000
10	MILLER	
20	SMITH	800
20	JONES	
20	SCOTT	
20	ADAMS	
20	FORD	
30	ALLEN	1600
30	WARD	1250
30	MARTIN	
30	BLAKE	
30	TURNER	
30	JAMES	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, sal,
2         lag(sal,1)
3         over ( partition by deptno
4               order by empno) as prev_sal
5 from emp
6 order by deptno, empno;

```

DEPTNO	ENAME	SAL	PREV_SAL
10	CLARK		
10	KING	5000	
10	MILLER		5000
20	SMITH	800	
20	JONES		800
20	SCOTT		
20	ADAMS		
20	FORD		
30	ALLEN	1600	
30	WARD	1250	1600
30	MARTIN		1250
30	BLAKE		
30	TURNER		
30	JAMES		

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```

SQL> select deptno, ename, sal,
2         lag(sal,1) IGNORE NULLS
3         over ( partition by deptno
4               order by empno) as prev_sal
5 from emp
6 order by deptno, empno;

```

DEPTNO	ENAME	SAL	PREV_SAL
10	CLARK		
10	KING	5000	
10	MILLER		5000
20	SMITH	800	
20	JONES		800
20	SCOTT		800
20	ADAMS		800
20	FORD		800
30	ALLEN	1600	
30	WARD	1250	1600
30	MARTIN		1250
30	BLAKE		1250
30	TURNER		1250
30	JAMES		1250

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

and then....

*“imagination is more  
important than knowledge”*

*- Albert Einstein*

more than just analytics

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

121

classical problems  
*made simple*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

122



*"remove the duplicates"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

123

```
SQL> select * from BAD_EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
7788	SCOTT	ANALYST	7566	09-DEC-82	3000
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600
7369	SMITH	CLERK	7902	17-DEC-80	800
7839	KING	PRESIDENT		17-NOV-81	5000
7566	JONES	MANAGER	7839	02-APR-81	2975
7876	ADAMS	CLERK	7788	12-JAN-83	1100
7844	TURNER	SALESMAN	7698	08-SEP-81	1500
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600
7369	SMITH	CLERK	7902	17-DEC-80	800
7902	FORD	ANALYST	7566	03-DEC-81	3000
7900	JAMES	CLERK	7698	03-DEC-81	950
7521	WARD	SALESMAN	7698	22-FEB-81	1250
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250
7934	MILLER	CLERK	7782	23-JAN-82	1300
7698	BLAKE	MANAGER	7839	01-MAY-81	2850
7782	CLARK	MANAGER	7839	09-JUN-81	2450

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

124

```
SQL> select * from BAD_EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
-----					
7788	SCOTT	ANALYST	7566	09-DEC-82	3000
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600
7369	SMITH	CLERK	7902	17-DEC-80	800
7839	KING	PRESIDENT		17-NOV-81	5000
7566	JONES	MANAGER	7839	02-APR-81	2975
7876	ADAMS	CLERK	7788	12-JAN-83	1100
7844	TURNER	SALESMAN	7698	08-SEP-81	1500
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600
7369	SMITH	CLERK	7902	17-DEC-80	800
7902	FORD	ANALYST	7566	03-DEC-81	3000
7900	JAMES	CLERK	7698	03-DEC-81	950
7521	WARD	SALESMAN	7698	22-FEB-81	1250
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250
7934	MILLER	CLERK	7782	23-JAN-82	1300
7698	BLAKE	MANAGER	7839	01-MAY-81	2850
7782	CLARK	MANAGER	7839	09-JUN-81	2450

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

125

```
SQL> select empno, rowid,
2         row_number() over
3         ( partition by empno order by rowid ) as r
4 from BAD_EMP
5 /
```

EMPNO	ROWID	R
-----		
7369	AAARXwAAEAAATlMAAA	1
7369	AAARXwAAEAAATlOAAA	2
7499	AAARXwAAEAAATlMAAB	1
7499	AAARXwAAEAAATlOAA B	2
7521	AAARXwAAEAAATlMAAC	1
7566	AAARXwAAEAAATlMAAD	1
7654	AAARXwAAEAAATlMAAE	1
7698	AAARXwAAEAAATlMAAF	1
7782	AAARXwAAEAAATlMAAG	1

...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

126

```
SQL> delete from BAD_EMP
2  where ROWID in
3  ( select rowid
4    from
5      ( select rowid,
6            row_number() over
7              ( partition by empno
8                order by rowid) as r
9          from BAD_EMP
10        )
11    where r > 1
12  )
13  /
```

2 rows deleted.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

127

*“5 highest salaries  
from each department”*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

128

```
SQL> select deptno, salary
  2   from
  3   ( select
  4       deptno,
  5       salary,
  6       rank() over (
  7         partition by deptno
  8         order by salary) top_5
  9   from EMPLOYEES
 10 )
 11 where top_5 <= 5
```

*“mind the gap”*

```
SQL> select X from T;
```

```

      X
-----
      2
      3
      4
      7
      8
     12
     13
     15
     16
     17
     19
     20

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

131

```
SQL> select X from T;
```

```

      X
-----
      2
      3
      4
      7
      8
     12
     13
     15
     16
     17
     19
     20

```

} 2-4  
 } 7-8  
 } 12-13  
 } 15-17  
 } 19-20

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

131

```
SQL> select
2     x,
3     lag(x) over ( order by x) prev
4 from T ;
```

X	PREV
2	
3	2
4	3
7	4
8	7
12	8
13	12
15	13
16	15
17	16
19	17
20	19

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

132

```
SQL> select
2     x,
3     lag(x) over ( order by x) prev
4 from T ;
```

X	PREV
2	
3	2
4	3
7	4
8	7
12	8
13	12
15	13
16	15
17	16
19	17
20	19

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

133

```
SQL> select
2      x,
3      lag(x) over ( order by x) prev
4  from T ;
```

X	PREV
2	
3	2
4	3
7	4
8	7
12	8
13	12
15	13
16	15
17	16
19	17
20	19

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

134

```
SQL> select
2      x,
3      lag(x) over ( order by x) prev
4  from T ;
```

X	PREV
2	
3	2
4	3
7	4
8	7
12	8
13	12
15	13
16	15
17	16
19	17
20	19

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

135

```

SQL> select
2     x,
3     case
4         when nvl(lag(x) over (order by x),x) != x-1
5             then x end loval
6 from t;

```

X	LOVAL
2	2
3	
4	
7	7
8	
12	12
13	
15	15
16	
17	
19	19
20	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

136

```

SQL> select
2     x,
3     case
4         when nvl(lag(x) over (order by x),x) != x-1
5             then x end loval
6 from t;

```

X	LOVAL
2	2
3	
4	
7	7
8	
12	12
13	
15	15
16	
17	
19	19
20	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

137



```

SQL> select
2     x,
3     case
4         when nvl(lag(x) over (order by x),x) != x-1
5             then x end loval
6 from t;

```

X	LOVAL
2	2
3	
4	
7	7
8	
12	12
13	
15	15
16	
17	
19	19
20	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

138

```

SQL> select
2     x,
3     case
4         when nvl(lag(x) over (order by x),x) != x-1
5             then x end loval
6 from t;

```

X	LOVAL
2	2
3	
4	
7	7
8	
12	12
13	
15	15
16	
17	
19	19
20	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

139

```

SQL> select x, max(loval) over (order by x) loval
2  from (
3  select x,
4         case
5             when nvl(lag(x) over (order by x),x) != x-1
6                 then x end loval
7  from t );

```

X	LOVAL
2	2
3	2
4	2
7	7
8	7
12	12
13	12
15	15
16	15
...	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

140

```

SQL> select x, max(loval) over (order by x) loval
2  from (
3  select x,
4         case
5             when nvl(lag(x) over (order by x),x) != x-1
6                 then x end loval
7  from t );

```

X	LOVAL
2	2
3	2
4	2
7	7
8	7
12	12
13	12
15	15
16	15
...	

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

141

```

SQL> select min(x) || '-' || max(x) ranges from (
2      select x,max(loval) over (order by x) loval
3      from (
4          select x,
5              case
6                  when nvl(lag(x) over (order by x),x) != x-1
7                      then x end loval
8          from t))
9      group by loval;

```

RANGES

```

-----
2-4
7-8
12-13
15-17
19-20

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

142

another solution

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

143

```

SQL> select
2   min(x) || '-' ||
3   case when min (x) = max (x)
4         then min(x)
5         else max(x)
6   end rng
7 from
8   (select X
9        , row_number() over (order by X) rn
10   from t
11   )
12 group by x - rn
13 order by min(x) ;

```

RNG

```

-----
2-4
7-8
12-13
15-17
19-20

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

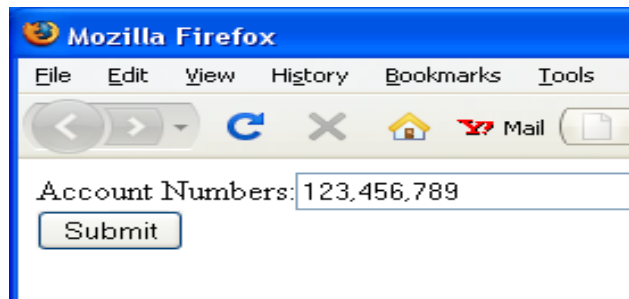
for the developers

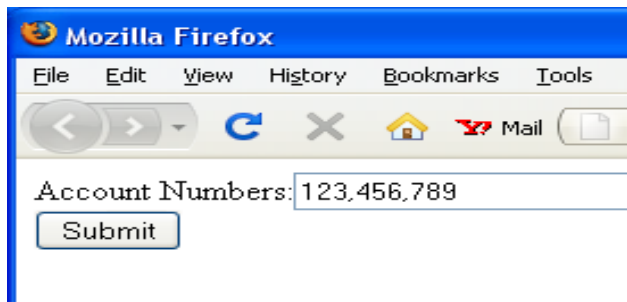
ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

145

## in-list processing





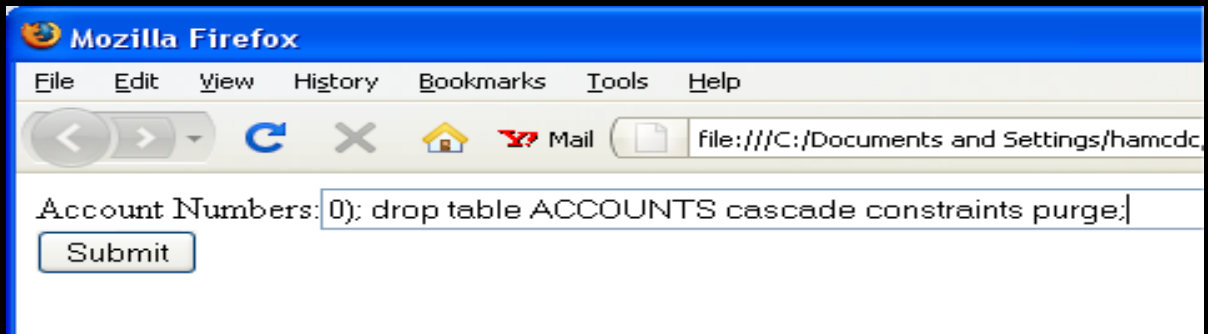
```
sql_string =  
    "select * from ACCOUNTS where ACCT_NO in ( " + :acct_input + ")"  
  
EXEC SQL PREPARE sql_string;
```

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

147





123,456,789

```
sql_string =  
  "select * from ACCOUNTS where ACCT_NO in ( :bindvar )"
```

```
EXEC SQL PREPARE sql_string;
```

ORA-01722: invalid number

```

SQL> exec :acct = '123,456,789'

SQL> select substr(:acct,
2         loc+1,nvl(
3         lead(loc) over ( order by loc ) - loc-1,
4         length(:acct)-loc)
5         ) list_as_rows
6 from (
7   select distinct (instr(:acct,',',1,level)) loc
8   from dual
9   connect by level < length(:acct)-
10      length(replace(:acct,','))+1
11  );

```

LIST\_AS\_ROWS

```

-----
123
456
789

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

150

```

SQL> with MY_LIST as ( select substr(:acct,
2         loc+1,nvl(
3         lead(loc) over ( order by loc ) - loc-1,
4         length(:acct)-loc)
5         ) val
6 from (
7   select distinct (instr(:acct,',',1,level)) loc
8   from dual
9   connect by level < length(:acct)-
10      length(replace(:acct,','))+1
11  )
12 select *
13 from ACCOUNTS
14 where ACCT_NO in ( select val from MY_LIST)

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

151



11.2+

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

152

opposite

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

153

listagg

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

154

classical problem

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

155

```
SQL> select deptno, ename
2   from   emp
3   order by 1,2;
```

DEPTNO	ENAME
10	CLARK
10	KING
10	MILLER
20	ADAMS
20	FORD
20	JONES
20	SCOTT
20	SMITH
30	ALLEN
30	BLAKE
30	JAMES
30	MARTIN
30	TURNER
30	WARD

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

156

## DEPTNO MEMBERS

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

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

157

```

SQL> select deptno , rtrim(ename,',') enames
2   from ( select deptno,ename,rn
3           from emp
4           model
5             partition by (deptno)
6             dimension by (
7               row_number() over
8                 (partition by deptno order by ename) rn
9             )
10          measures (cast(ename as varchar2(40)) ename)
11          rules
12            ( ename[any]
13              order by rn desc = ename[cv()]||', '||ename[cv()+1])
14          )
15   where rn = 1
16   order by deptno;

```

#### DEPTNO ENAMES

```

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

```

- Rob Van Wijk

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

158

```

SQL> select deptno,
2         substr(max(sys_connect_by_path(ename, ','), 2) members
3   from (select deptno, ename,
4             row_number ()
5             over (partition by deptno order by empno) rn
6   from emp)
7   start with rn = 1
8   connect by prior rn = rn - 1
9   and prior deptno = deptno
10  group by deptno
11  /

```

#### DEPTNO MEMBERS

```

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

```

- Anon

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

159

```

SQL> select deptno,
2      xmltransform
3      ( sys_xmlagg
4        ( sys_xmlgen(ename)
5          ),
6        xmltype
7        (
8          '<?xml version="1.0"?><xsl:stylesheet version="1.0"
9 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
10       <xsl:template match="/">
11         <xsl:for-each select="/ROWSET/ENAME">
12           <xsl:value-of select="text()" />;</xsl:for-each>
13         </xsl:template>
14       </xsl:stylesheet>'
15        )
16      ).getstringval() members
17 from emp
18 group by deptno;

```

DEPTNO MEMBERS

```

-----
10 CLARK;MILLER;KING;
20 SMITH;FORD;ADAMS;SCOTT;JONES;
30 ALLEN;JAMES;TURNER;BLAKE;MARTIN;WARD;

```

- Laurent Schneider

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

160

```

SQL> create or replace type string_agg_type as object
2  (
3    total varchar2(4000),
4
5    static function
6      ODCIAggregateInitialize(sctx IN OUT string_agg_type )
7      return number,
8
9    member function
10     ODCIAggregateIterate(self IN OUT string_agg_type ,
11                          value IN varchar2 )
12     return number,
13
14    member function
15     ODCIAggregateTerminate(self IN string_agg_type,
16                           returnValue OUT varchar2,
17                           flags IN number)
18     return number,
19
20    member function
21     ODCIAggregateMerge(self IN OUT string_agg_type,
22                        ctx2 IN string_agg_type)
23     return number
24 );

```

- Tom Kyte

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

161

# hard

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

162

```
SQL> select deptno,  
2         listagg( ename, ',' )  
3         within group (order by empno) members  
4   from   emp  
5  group  by deptno;
```

DEPTNO MEMBERS

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

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

163

other goodies

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

164

KEEP extension

ORACLE

165 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

165

"Show me lowest salary for each department..."



ORACLE

166 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

"Show me lowest salary for each department..."

```
SQL> select deptno, min(sal)
2   from emp
3  group by deptno;
```



ORACLE

166 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |



"Show me lowest salary for each department..."

```
SQL> select deptno, min(sal)
2   from emp
3  group by deptno;
```

"...and I need to know who has that lowest salary as well"



ORACLE

166 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

"Show me lowest salary for each department..."

```
SQL> select deptno, min(sal)
2   from emp
3  group by deptno;
```

"...and I need to know who has that lowest salary as well"

```
SQL> select deptno, empno, min(sal)
2   from emp
3  group by deptno;
```

ORA-00979: not a GROUP BY expression



ORACLE

166 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

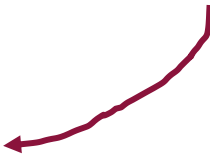
```

SQL> select deptno, min(sal), min(empno)
2      KEEP ( dense_rank FIRST order by sal) empno
3 from emp
4 group by deptno
5 /

```

DEPTNO	MIN (SAL)	EMPNO
10	1300	7934
20	800	7369
30	950	7900

Emp 7934 has the  
lowest salary in dept 10



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

167

# inverse analytics

ORACLE

168 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

168

recall

ORACLE

169 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

169

cume\_dist

ORACLE

170 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

170

```
SQL> select ename, sal,
2      100*cume_dist() over ( order by sal ) as pct
3  from emp
4  order by sal;
```

ENAME	SAL	PCT
SMITH	800	7.14
JAMES	950	14.29
ADAMS	1100	21.43
WARD	1250	35.71
MARTIN	1250	35.71
MILLER	1300	42.86
TURNER	1500	50.00
ALLEN	1600	57.14
CLARK	2450	64.29
BLAKE	2850	71.43
JONES	2975	78.57
FORD	3000	92.86
SCOTT	3000	92.86
KING	5000	100.00

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

171

*"what is the 60<sup>th</sup> percent salary ?"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

172

```
SQL> select ename, sal,
2      100*cume_dist() over ( order by sal ) as pct
3      from emp
4      order by sal;
```

ENAME	SAL	PCT
SMITH	800	7.14
JAMES	950	14.29
ADAMS	1100	21.43
WARD	1250	35.71
MARTIN	1250	35.71
MILLER	1300	42.86
TURNER	1500	50.00
ALLEN	1600	57.14
CLARK	2450	64.29
BLAKE	2850	71.43
JONES	2975	78.57
FORD	3000	92.86
SCOTT	3000	92.86
KING	5000	100.00

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

173

```
SQL> select ename, sal,
2      100*cume_dist() over ( order by sal ) as pct
3      from emp
4      order by sal;
```

ENAME	SAL	PCT
SMITH	800	7.14
JAMES	950	14.29
ADAMS	1100	21.43
WARD	1250	35.71
MARTIN	1250	35.71
MILLER	1300	42.86
TURNER	1500	50.00
ALLEN	1600	57.14
CLARK	2450	64.29
BLAKE	2850	71.43
JONES	2975	78.57
FORD	3000	92.86
SCOTT	3000	92.86
KING	5000	100.00

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

174

```

SQL> select
  2     percentile_disc(0.6)
  3         within group
  4         (order by sal) as dicrete_pct,
  5     percentile_cont(0.6)
  6         within group
  7         (order by sal) as continuous_pct
  8 from emp;

```

DICRETE_PCT	CONTINUOUS_PCT
-----	-----
2450	2280

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

175

classical problems  
*made simple*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

176

*"median salary for  
each department?"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

177

```
SQL> select deptno, avg(distinct sal) median
 2  from
 3      (select cp1.deptno, cp1.sal
 4         from emp cp1, emp cp2
 5         where cp1.deptno = cp2.deptno
 6         group by cp1.deptno, cp1.sal
 7         having sum(decode(cp1.sal, cp2.sal, 1, 0)) >=
 8                    abs(sum(sign(cp1.sal - cp2.sal))))
 9  group by deptno
10  /
```

DEPTNO	MEDIAN
10	3725
20	2975
30	1375
40	1300

*huh?*



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

178

```

SQL> select
2      deptno,
3      percentile_cont(0.5)
4      within group (order by sal) as median
5  from emp
6  group by deptno;

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

179

```

SQL> select
2      deptno,
3      percentile_cont(0.5)
4      within group (order by sal) as median
5  from emp
6  group by deptno;

```

```

SQL> select
2      deptno,
3      median(sal)
4  from emp
5  group by deptno;

```

DEPTNO	MEDIAN(SAL)
10	3725
20	2975
30	1375
40	1300

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

179



## hypothetical analytics

ORACLE

180 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

180

*"if I was paid \$3000,  
where would I rank in  
each department?"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

181

```
SQL> select
  2     deptno,
  3     rank(3000) within group
  4       ( order by sal ) as ranking
  5 from emp
  6 group by deptno;
```

DEPTNO	RANKING
10	2
20	4
30	7
40	2

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

182

ratio\_to\_report

ORACLE

183 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

183

## *"salary percentage breakdown across employees"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

184

```
SQL> select
  2      empno,
  3      ename,
  4      sal,
  5      100*ratio_to_report(sal) over () as pct
  6 from emp;
```

EMPNO	ENAME	SAL	PCT
7521	WARD	1250	4.69
7566	JONES	2975	11.17
7654	MARTIN	1250	4.69
7698	BLAKE	2850	10.70
7782	CLARK	2450	9.20
7788	SCOTT	3000	11.27
7839	KING	5000	18.78
7844	TURNER	1500	5.63
7876	ADAMS	1100	4.13
7900	JAMES	950	3.57
7902	FORD	3000	11.27

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

185

a couple more things ...

*not really analytic ?*

ORACLE

186 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

1) partitioned outer join

ORACLE

187 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select * from bookings;
```

HR	ROOM	WHO
8	Room2	PETE
9	Room1	JOHN
11	Room1	MIKE
14	Room2	JILL
15	Room2	JANE
16	Room1	SAM

```
SQL> select * from hrs;
```

HR
8
9
10
11
12
13
14
15
16

ORACLE

188 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# bookings by hour

## (conventional outer join)

ORACLE

189 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

189

```
SQL> SELECT hrs.hr, t1.room, t1.who
2  from   hrs, bookings t1
3  where  hrs.hr = t1.hr(+)
```

HR	ROOM	WHO
8	Room2	PETE
9	Room1	JOHN
10		
11	Room1	MIKE
12		
13		
14	Room2	JILL
15	Room2	JANE
16	Room1	SAM

ORACLE

190 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

190

# room occupancy

(*partitioned* outer join)

ORACLE

191 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

191

```
SQL> SELECT hrs.hr, t1.room, t1.who
2 FROM   bookings t1
3 PARTITION BY (t1.room)
4 RIGHT OUTER JOIN hrs ON (hrs.hr = t1.hr);
```

HR	ROOM	WHO
8	Room1	
9	Room1	JOHN
10	Room1	
11	Room1	MIKE
12	Room1	
13	Room1	
14	Room1	
15	Room1	
16	Room1	SAM
8	Room2	PETE
9	Room2	
10	Room2	
11	Room2	
12	Room2	
13	Room2	
14	Room2	JILL
15	Room2	JANE
16	Room2	

ORACLE

192 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

192

## 2) width\_bucket

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

193

```

SQL> select
2   object_id,
3   width_bucket(object_id,
4       1000,
5       90000,
6       10) bucket
7   from dba_objects

```

ORACLE

194 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

194

```

SQL> select
2   object_id,
3   width_bucket(object_id,
4       1000,
5       90000,
6       10) bucket
7   from dba_objects

```

OBJECT_ID	BUCKET
913	0
...	
3231	1
...	
5858	1
...	
14920	2
...	
42421	5
...	
91635	11



1 .. buckets

ORACLE

194 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

194



```
SQL> select
2   object_id,
3   width_bucket(object_id,
4       1000,
5       90000,
6       10) bucket
7   from dba_objects
```

OBJECT_ID	BUCKET
913	0
...	
3231	1
...	
5858	1
...	
14920	2
...	
42421	5
...	
91635	11

&lt; min

1 .. buckets

ORACLE

194 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

194

```
SQL> select
2   object_id,
3   width_bucket(object_id,
4       1000,
5       90000,
6       10) bucket
7   from dba_objects
```

OBJECT_ID	BUCKET
913	0
...	
3231	1
...	
5858	1
...	
14920	2
...	
42421	5
...	
91635	11

&lt; min

1 .. buckets

&gt; max

ORACLE

194 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

194

## back to analytics

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

195

## things to note

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

196

cannot be a predicate

```
SQL> select ename, deptno, sal
  2   from emp
  3   where
  4       sum(sal) over
  5         ( partition by deptno) > 10;
sum(sal) over
*
```

ERROR at line 4:  
ORA-00934: group function is not allowed here

## inline view

```
SQL> select ename, deptno, sal
2  from (
3      select ename, deptno, sal,
4          sum(sal) over
5              ( partition by deptno) as deptsal
6      from emp
7      )
8  where deptsal > 10;
```

careful with views...

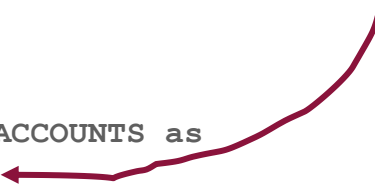
ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

201

indexed column

```
create view RANKED_ACCOUNTS as
select account_num,
       customer_name,
       acct_type_code,
       rank() over ( order by gross_sales ) as seq
from   ACCOUNTS;
```



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

202

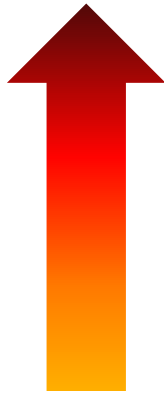
```
SQL> select * from RANKED_ACCOUNTS
      2 where ACCOUNT_NUM = 12345
```

Id	Operation	Name
0	SELECT STATEMENT	
1	VIEW	RANKED_ACCOUNTS
2	WINDOW SORT	
3	TABLE ACCESS FULL	ACCOUNTS

a more holistic view

*(part 1)*

question



solution

ORACLE

205

recall

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

206

```
SQL> select deptno,  
2         listagg(ename, ',')  
3         within group (order by empno)  
4   from emp  
5   group by deptno;
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

207

still challenges

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

208



real example

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

209



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

210

# AML

*anti money laundering*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

211

*"Find 10 consecutive deposits in a  
24 hour period, then  
a withdrawal within three days of  
the last deposit, at a different outlet"*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

212

ACCT	TSTAMP	WTHD_TSTAMP	T	AMT
54261	25/01/13 17:20:55		D	100
54261	25/01/13 17:56:58		D	165
54261	26/01/13 11:24:14		D	30
54261	26/01/13 11:47:53		D	45
54261	26/01/13 12:59:38		D	100
54261	26/01/13 13:26:04		D	80
54261	26/01/13 14:41:09		D	50
54261	26/01/13 14:53:12		D	50
54261	26/01/13 15:15:05		D	50
54261	26/01/13 15:51:17		D	50
54261	26/01/13 16:15:02		D	120
54261	26/01/13 16:36:51		D	100
54261	26/01/13 16:55:09		D	100
54261	26/01/13 18:07:17	26/01/13 18:07:17	W	-500

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

213

hard...

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

214

"analysis"

≈

"trends" / "patterns"

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

215

12c

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

216

# pattern matching

```

SQL> select acct, tstamp, withd_tstamp, txn_type, amt
  2  from account_txns
  3  MATCH_RECOGNIZE
  4  (
  5    partition by acct
  6    order by tstamp
  7    measures
  8      dep.tstamp dep_tstamp,
  9      withd.tstamp withd_tstamp
 10
 11    all rows per match
 12    pattern ( dep{10,} withd )
 13    define
 14      dep as
 15        txn_type = 'D',
 16      withd as
 17        txn_type = 'W'
 18        and last(dep.tstamp)-first(dep.tstamp) < interval '1' day
 19        and withd.tstamp - last(dep.tstamp) < interval '3' day
 20        and withd.location != last(dep.location)
 21  )

```

a more holistic view

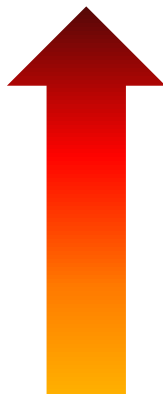
*(part 2)*

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

219

question



solution

ORACLE

220

# frequent itemsets



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

221

```
SQL> select DEMO  
       2 from DEMOGRAPHIC;
```

DEMO

-----

Child

Teenager

Adult



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select DEMO
      2 from DEMOGRAPHIC;
```

```
SQL> select SEX
      2 from GENDER;
```

```
Chi SEX
Tee -----
Adu Male
      Female
      Unspecified
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

```
SQL> select DEMO
      2 from DEMOGRAPHIC;
```

```
SQL> select SEX
      2 from GENDER;
```

```
SQL> select CEREAL_NAME
      2 from BRANDS;
```

```
Chi SEX
Tee -----
Adu Male
      CEREAL_NAME
      -----
      Cheerios
      CinamonCrunch
      HoneyNutCheerios
      FrootLoops
      FrostedFlakes
      SpecialK
      LuckyCharms
      CocoPuffs
      FrostedMiniWheats
      RiceKrispies
```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |



```
SQL> desc BREAKFAST_FOOD
```

Name	Null?	Type
CUST_ID		NUMBER (38)
DEMOGRAPHIC		VARCHAR2 (20)
GENDER		VARCHAR2 (20)
BRAND		VARCHAR2 (20)
RATING		VARCHAR2 (20)

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

223

```
SQL> select *
2 from BREAKFAST_FOOD;
```

CUST_ID	DEMOGRAPHIC	GENDER	BRAND	RATING
1	Child	Male	Wheaties	Hate
2	Teenager	Female	RaisinBran	Hate
3	Child	Male	FrootLoops	Love
4	Adult	Female	RaisinBran	Love
5	Child	Male	Wheaties	Hate
6	Adult	Male	FrootLoops	Love
7	Child	Female	SultanaBran	Hate
8	Child	Female	Special K	Hate
9	Teenager	Male	Special K	Hate
10	Child	Female	CinamonCrunch	OK
11	Teenager	Male	RiceKrispies	OK
12	Child	Male	CornFlakes	Hate

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

224

```
SQL> create view CUSTOMER_ATTRIBUTES as
  2  SELECT cust_id, demographic
  3  FROM   breakfast_food
  4  UNION ALL
  5  SELECT cust_id, brand
  6  FROM   breakfast_food
  7  UNION ALL
  8  SELECT cust_id, rating
  9  FROM   breakfast_food
 10  UNION ALL
 11  SELECT cust_id, gender
 12  FROM   breakfast_food
 13  /
```

View created.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

225

```
SQL> create or replace type VC_LIST
  2  as table of varchar2(30);
  3  /
```

Type created.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

226

```

SQL> SELECT
  2   CAST (itemset as vc_list) itemset
  3   ,round(100*support/total_tranx,2) pct
  4 FROM
  5   TABLE(DBMS_FREQUENT_ITEMSET.FI_TRANSACTIONAL
  6            ( cursor
  7              ( SELECT * FROM CUSTOMER_ATTRIBUTES )
  8              , 0.03 -- threshold
  9              , 3
 10              , 4
 11              , cursor
 12                (SELECT cereal_name FROM brands)
 13              , NULL
 14              )
 15            )
 16 order by 2 desc
 17 /

```

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

227

ITEMSET	PCT
VC_LIST('RaisinBran', 'Child', 'Hate')	6.27
VC_LIST('RaisinBran', 'Hate', 'Male')	4.65
VC_LIST('Child', 'Hate', 'JustRight')	4.40
VC_LIST('CornFlakes', 'Hate', 'Male')	4.27
VC_LIST('AllBran', 'Child', 'Male')	4.20
VC_LIST('Child', 'Hate', 'SultanaBran')	4.15
VC_LIST('Adult', 'FrootLoops', 'Hate')	4.12
VC_LIST('Child', 'CornFlakes', 'Hate')	4.06
VC_LIST('Child', 'Hate', 'Wheaties')	4.04
VC_LIST('Adult', 'CocoPuffs', 'Hate')	3.81
VC_LIST('RaisinBran', 'Child', 'Hate', 'Male')	3.78
VC_LIST('Adult', 'RaisinBran', 'Love')	3.75

[snip]

28 rows selected.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

228

*“what will my child like  
for breakfast that is not CocoPuffs”*

```
SQL> SELECT CAST (itemset as vc_list) itemset
  2 FROM
  3   table( DBMS_FREQUENT_ITEMSET.FI_TRANSACTIONAL
  4           ( cursor
  5             ( SELECT * FROM CUSTOMER_ATTRIBUTES )
  6             , 0.03 -- threshold
  7             , 3
  8             , 4
  9             , cursor
 10             (SELECT cereal_name FROM brands)
 11             , NULL))
 12 where 'Child' member of itemset
 13 and 'CocoPuffs' not member of itemset
 14 and 'Love' member of itemset
 15 /
```

ITEMSET

VC\_LIST('Child', 'FrootLoops', 'Love')

wrap up

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

analytics

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

232

cool

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

233

less SQL

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

234

easier to read SQL

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

235

Connor McDonald

@connor\_mc\_d

ORACLE

236 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

# ORA-03113

@connor\_mc\_d

ORACLE

236 Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Creative Commons Image Resources

[https://commons.wikimedia.org/wiki/File:Horse\\_Race\\_Finish\\_Line\\_\(11888565543\).jpg](https://commons.wikimedia.org/wiki/File:Horse_Race_Finish_Line_(11888565543).jpg)

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |